11

Modelling Process and Supply Chain Scheduling Using Hybrid Meta-heuristics

Soumya Banerjee[1], G.S.Dangayach[2], S.K.Mukherjee[3], and P.K. Mohanti[4]

[1] Dept. of Computer Science and Engineering, Birla Institute of Technology, Mesra, India soumyabanerjee@bitmesra.ac.in
[2] Department of Mechanical Eng., Malviya National Institute of Technology, Jaipur, India
[3] Vice Chancellor, Birla Institute of Technology, Mesra, India
[4] University of New Brunswick, New Brunswick, Canada

Summary. This chapter proposes a natural stigmergic computational technique *Bee Colony* for process scheduling and optimization problems developed by mimicking social insects' behavior. The case study considered in the chapter is a milk production center, where process scheduling, supply chain network etc. are crucial, as slight deviation in scheduling may lead to perish out the item causing financial loss of the plant. The process scheduling of such plants extensively deals with multi-objective conflicting criteria, hence the concept of *Pareto Dominance* has been introduced in the form of *Pareto Bee Colony Optimization*. Some facts about social insects namely bees are presented with an emphasis on how they could interact and self organized for solving real world problems. Finally, a performance simulation and comparison has been accomplished envisaging other similar bio-inspired algorithms.

Key words: Supply Networks, Scheduling, Bee Colony Optimization, Multi-objective Optimization, Pareto Bee Colony.

11.1 Introduction and Background

Supply networks are organizations of partially autonomous production and distribution centers through which goods are processed and delivered to customers. Optimizing the activities of a supply network to improve production throughput and timeliness of the delivery requires dealing with a number of large-scale, interrelated assignments, scheduling and routing problems. The optimization is especially challenging for a supply network that delivers rapidly perishable goods, such as raw materials used for manufacturing foods and beverages. The perishable goods are only used within a period of restricted time limit, so it is expected that their production and delivery are made only on stipulated demand and even their routing through proper

S. Banerjee et al.: *Modelling Process and Supply Chain Scheduling Using Hybrid Meta-heuristics*, Studies in Computational Intelligence (SCI) **128**, 277–300 (2008)
www.springerlink.com

channel also becomes an issue. Naturally the problem of such type requires multi optimization with large number of constraints at different stages. The specialty of such optimization is not only about their costs but also flexibility and robustness of the solution could also be considered. It is observed that any small deviation over local activity may inject a cascaded delay while deploying the common resources. Therefore, referring the aspect of perishable material for food industry, this may turn into substantially significant financial loss and even degrading the brand value of the company, and all the conventional predefined optimal solutions could become impractical due to various other constraints. Professionals find it as trade off between the minimizing the costs of operations and providing shock absorption over these disruptions.

There are certain commercial tools (like CPLEX) available to address these issues. A considerable number of works that demonstrate the approach of supply network logistics issues, but comparatively less versions of multi optimization scheduling based on meta-heuristics have been established. We propose a novel hybrid meta-heuristics optimally in a supply network for perishable material concerned with food industry. The genesis of the search strategy is based on evolutionary computation method, mainly to exploit its efficient exploration/exploitation capability in the large search space of the main decision variables characterizing our scheduling problem. The solution integrates the followings:

- A detailed mathematical model of the logistic problem that unambiguously specifies the free decision variables.
- A set of fast heuristics organized in a hierarchical structure that is able to construct a fully feasible solution starting from an initial assignment of a subset of decision variables.
- A multi-objective bee colony based rough set algorithm that searches for the set of best tradeoff solutions considering both the costs and the robustness of the corresponding schedules.

The proposed solution is presented in the context of reducing the risk and uncertainty in optimization. The optimization algorithm returns a set of solutions with different cost and risk tradeoffs, allowing the analyst to adapt the planning depending on the attitude to risk. The subsequent section presents an overview of supply chain modelling through diversified methodologies especially envisaging adaptive and intelligent techniques.

Background

The globalization, dynamics, and frequent variations of customer demands on today's markets increase the needs of companies to form supply chains (SCs) and cooperative business partnerships that enable them to survive on today's competitive market [6, 7]. SCs are networks of autonomous business entities that collectively procure, manufacture, and distribute certain products. The objective of a SC is to respond efficiently to customer demands and at the same time, it must minimize the cost of all participating business entities. To achieve this objective the supply chain management (SCM) system must coordinate and optimize the procurement, production, and distribution of goods. In reality, however, SCs are often operating in dynamic and non-homogenous cultural environments. Therefore, current SCM systems need to adopt adaptive learning features and reasoning of theory of evidence to reflect the changes in the dynamic cultural environment. In practical research, there are couples of supply chain works primarily concerned with adaptive reasoning and seems to be hybrid intelligent. For example, Several systems were developed to model the SCs complexity using a GA, for examples Truong and Azadivar [8] have integrated GAs, mixed integer programming methods, and simulation techniques into a hybrid optimization model, while other researchers use GAs and Pareto Optimal techniques [9]. Furthermore, Al-Mutawah, Lee, and Cheung have developed a Distributed Multi-objective Genetic Algorithm (DMOGA) to solve the SC optimization problem in [10]. One common limitation of DMOGA and other typical genetic based implementation of multi-objective optimization is the inheritance process of GA, which restricts the parents to transfer experiences only to their offspring, ignoring the influence of other external sources. In real world, particularly in a distributed environment, SC applications data are collected from heterogeneous sources, implying the need to co-opt other sources of influence as well.

11.2 Related Works on Meta-heuristics

As it has been discussed in previous sections, the supply chain is a complex network of facilities and organizations with interconnected activities but different and conflicting objectives. Many companies are interested in analyzing their supply chain as an entire and unique system to be able to improve their business. However, in most cases the task of designing, analyzing and managing the supply chain has been done based on experience and intuition; very few analytical models and design tools have been used in the process. This implies that finding the best supply chain strategies for a particular firm, group of firms or sector poses significant challenges to the industry and academia. The optimization literature focuses on algorithms for computing solutions to constrained optimization problems.

Meta-heuristics have many desirable features to be an excellent method to solve very complex SCM problems: in general they are simple, easy to implement, robust and have been proven highly effective to solve hard problems. Several other aspects are worth to mention. The first one is the meta-heuristicsc modular nature that leads to short development times and updates, given a clear advantage over other techniques for industrial applications. This modular aspect is especially important given the current times of implementing a Decision Support System (DSS) in a firm and the rapid changes that occurs in the area of SCM. The next important aspect is the amount of data involved in any optimization model for an integrated supply chain problem, which can be overwhelming. The complexity of the models for the SCM and the incapacity of solving in real time some of them by the traditional techniques, force the use of the obvious technique to reduce this complex issue by data aggregation [11]. However this approach can hide important aspects that impact the decisions. Other reports published presented a complex vehicle routing model to the distribution in the food and beverages industries [12].

A literature search in food science and technology databases reveals that optimization using response surface modelling (RSM) has been, and continues to be, the most common approach. RSM techniques were introduced in the 1950s associated with design of experiments methods [1, 2]. Although the usefulness of RSM in certain conditions must be recognized, this approach has a number of important drawbacks due to the empirical, local and stationary nature of the simple algebraic models used. In contrast, a number of powerful model-based optimization methods have been developed during the last decades, which use more rigorous, time-dependent models. Primarily, at the core is the problem domain from which instances were drawn. The problem domain consist twenty four distinct instance classes on the basis of twelve distinct problem specifications. Except for the 100-job single machine total weighted tardiness instance class, for each instance class, a new benchmark set of 125 instances was created. Furthermore, solutions were obtained for all 3000 instances and recorded to serve as reference for future research. In order to find these solutions, new solution representations were developed for the problems with a parallel machine environment. Moreover, well known speed up techniques for the single machine total weighted tardiness problem were adapted to the constraints posed by objective functions that ignore weights or impose a unit penalty on each late job. Finally, the speed up techniques was adapted to work in machine environments with more than one machine in parallel [3].

Mixed planning and scheduling problem was discussed in length and it has been shown how to extend a conventional scheduler by some planning capabilities during the investigation on complex process models [4]. Balanced theory and practice of planning and scheduling in supply chains also have become prominent area of implementation [5]. The project first gives an overview of the various planning and scheduling models that have been studied in the literature, including lot sizing models and machine scheduling models. It subsequently categorizes the various industrial sectors in which planning and scheduling in the supply chains are important; these industries include continuous manufacturing as well as discrete manufacturing followed by the description how planning and scheduling models can be used in the design and the development of decision support systems for planning and scheduling in supply chains and discuss in detail the implementation of such a system at the Carlsberg A/S beer-brewer in Denmark.

The trend of research and implementation found to be more practical when natural heuristics has been incepted in scheduling the activities. Many project tasks and manufacturing processes consist of interdependent time-related activities that can be represented as networks. Deciding which of these sub-processes should receive extra resources to speed up the whole network (i.e. where activity crashing should be applied) usually involves the pursuit of multiple objectives amid a lack of a *priori* preference information. A common decision support approach lies in first determining efficient combinations of activity crashing measures and then pursuing an interactive exploration of this space.

11.3 Motivation and Importance Behind the Model

Most of the raw material of food and beverages are perishable due to their complex preservation approach and thus conventional *Just in Time (JIT)* methodology doesn't hold good as it seldom fails in uncertainty and ambiguity. The distribution network of finished frozen food products also requires utmost prompt delivery over a broader geographical coverage preventing them from damage in terms of food value and nutrition. The vehicle used for dispatching have limited capacity, and so large demands require several vehicle loads to transport all the products at proper places in time. These activities have to be properly synchronized, because the unloading at the customer site must be continuous in order to prevent compromising the food value properties of the product. Each production centers of food processing aims to increase resource utilization decrease costs and ensure the timeliness of the deliveries. Hence, those centers pursue multiple, contradictory goals. At present, many companies tend either to rely on skilled operators that work out production plans based on their experience, or to plan production operations on very short time horizons, sacrificing the optimization on longer horizon to achieve a reduced risk of delayed delivery.

The plethora of different type ambiguous problems of supply network envisages the present working project deploying certain novel components of evolutionary computations. The aim is to present a smarter and easily understood model, which could assist the logistics managers of food industry for scheduling their supply network in an optimized direction both on and off the production (including distribution).

Concept of Hybrid Meta Heuristics for the Proposed Model

We consider the problem of scheduling of events in the form of Directed Acyclic Graph (DAG). Each node in the graph represents an executable task. Each directed edge represents a precedence constraint (or simply dependence) between two tasks; the sink node cannot start execution until the source node has finished and the transmission of the required amount of data from the source node to the sink node has been completed. We assume that the DAG has always a single entry node (i.e. a node with no parents) and a single exit node (i.e. a node with no children). The target environment consists of a set of heterogeneous events, which are fully connected; a data transfer cost is given for each pair of events (like controlling the procurement of more perishable raw materials, enabling the dispatch of food product and beverage over the distribution network, etc). A task can execute on any available events; the execution cost of each task on each event is also given. The task scheduling problem is to allocate tasks for execution onto events in such a way that precedence constraints are respected and the overall execution time is minimized. It is assumed that only one task can execute on an event at a time and once a task has started execution it cannot be preempted.

The heuristic consists of three phases: (a) Ranking; (b) Group creation; and (c) Scheduling independent tasks within each group. For each of the stages the proposal is combining two different techniques concerned to tackle both optimization and uncertainty in the execution of the events. We incorporate Bee colony optimization (BCO) and rough set approach for this mode.

In the first phase, a weight is assigned to each node and edge of the graph; this is based on averaging all possible values for the cost of each node (or edge, respectively) on each events (or combination of events, respectively). Using this weight, upward ranking is computed and a rank value is assigned to each node. The rank value, of a node r_i is recursively defined as follows:

$$ri = wi + max(c_{ij+rj}), \tag{11.1}$$

where $\forall j \in s_i$ and w_i is the weight of node i, S_i is the set of immediate successors of node i and c_{ij} is the weight of the edge connecting nodes i and j.

In the second phase, nodes are sorted in descending order of their rank value; using this order, they are considered for assignment to groups as follows. The first node (i.e. the node with the highest rank value) is added to a group numbered 0. Successive nodes, always in descending order of their rank value, are placed in the same group as long as they are independent with all the nodes already assigned to the group (i.e. there is no dependence between them in the DAG). If dependence is found, then the node with the smallest rank value (i.e. the sink of the dependence) is made the member of a new group; the new group's number is the current group's number increased by one. Again, subsequent tasks, in terms of their rank value, will be added to this group as long as they are not dependent to any other node which is a member of this group; if they are, a new group will be created and so on. The outcome from this process is a set of ordered groups, each of which consists of a number of *independent* tasks, and has a predetermined priority (based on the original ranking of the nodes; a smaller group number indicates higher priority).

The third phase of heuristics comprises of a schedule of the DAG can be obtained by considering each group in ascending order of its number, and using *any* heuristic for scheduling the independent tasks within each group. It is noted that the input of the latter heuristic will be a set of (independent) tasks; a set of machines; the array giving the execution cost of each node on any machine; and, another array giving the earliest time that each task may start execution on each event.

11.4 Bee Colony Optimization

Various unsocial insect colonies such as ants, wasps, termites and bees exhibit remarkable problem solving behavior. Although a single insect is quite limited in its ability, complex behavior is exhibited at the level of the colony that emerges from the interactions of the individual insects [13]. This phenomenon is called Self-Organization. The foraging behavior of honey bees has been extensively studied and is a useful example of self-organization. Computational biology and modelling of these self organized properties mediated in solving plenty of complex optimization and scheduling problems.

Mathematical Model of Foraging for Honey Bees

Foraging is an interesting property to observe for honey bees and it is complex process involving large number of individuals collecting food from many different sources. Differential equation models have shown how quite simple communication mechanisms can produce complex and functional group level foraging patterns. Here, we concentrate our focus on the mathematical aspects of foraging including waggle dance of bees during the foraging. For example, although individual honey bee foragers follow only a small number of the waggle dances advertising flower patches, the colony can nonetheless focus its foraging effort on the most profitable patches. Similarly, certain ants deploy their foragers preferentially on the shorter of two paths, despite few if any individual insects directly comparing the paths [15,16]. The potential benefit of the existing mathematical models are to understand how population change through time. The number of bees foraging for a particular food source can be represented as single variable that changes its value as the insects are recruited to and abandon the source. These recruitment and abandonment rates can be written as functions of the number of insects foraging at a source, waiting at the nest, or scouting for new sources. There are many impressive literatures available on the different aspects of differential equation based modelling of insects encompassing the foraging [17–26].

In the mathematical model, several behavioral states could be contemplated. Colonies have access to n number of food sources. Each state has an associated variable, indexed by source (by default). Hence, the different sates in the dynamic model would like to be:

- Waiting (denoted as W): Waiting at the nest and available to start foraging.

- Searching (denoted as S): Searching for food sources.
- Exploiting (E_i): Exploiting food source i. Workers in this state do not directly recruit nest mates, although they may leave signals, such as pheromone trails, that increase the likelihood of other foragers finding the source.
- Recruiting (R_i): Attempting to recruit nest mates to food source i.
- Following (F_i): Attempting to follow recruiters to food source i.

In order to model and deploy the differential equation in these behavioral states of bees, we would like to establish a series of mathematical anomalies stated as follows:

- A waiting worker W can become an exploiting forager at source i E_i through three different routes, iff she might be activated to search (through function a), and then discover the food source (through d_i). Or, she might be led toward the food source through direct contact or communication (h_i) with another worker, arriving (s_i) only if the communication is successful. Finally, she might reach a food source by following an indirect signal, such as a pheromone trail (through j_i).
- The function j_i represents indirect recruitment, where successful foragers influence their environment in a manner that increases the chance of nest mates finding the food.
- The function f_i represents direct recruitment, where successful foragers either physically lead nest mates to the food source or directly communicate, in the nest, the location of the source [27].
- The population of workers in the nest, W, increases as searchers are deactivated (b), exploiters retire from foraging (g_i), and followers get lost and return to the nest (v_i). Conversely, the population decreases as nest workers are activated to search (a), as they are led by indirect recruitment signals to become exploiters (j_i), and as they begin to follow direct recruitment to various food sources (h_i).

Considering these dynamic conditions, we can write series of equations [14]:

$$dw/dt = b + \sum_{i=1}^{n} g_i + \sum_{i=1}^{n} v_i - a - \sum_{i=1}^{n} j_i - \sum_{i=1}^{n} h_i \tag{11.2}$$

where b is deactivated searchers as population of workers w increases in the nest, g_i denotes exploiters retire from foraging, v_i denotes return to the nest, a denotes activated workers, j_i is the recruitment signal and h_i denotes the direct recruitment of different food source.

The exhaustive mathematical treatment even prescribe to estimate the optimal investment in workers by colonies that use this foraging mechanism. A productivity function can be defined as to show how foraging efficiency depends on the maximum number of ants foraging at a food source at stable equilibrium [28]. These collective decision making of bees assist to model more complex situation very similar to group of robots taking a decisions in a group.

11.5 Multi-Objective Optimization and Standard Bee Colony Optimization Algorithm

Most real world optimization problems are naturally posed as multi-objective optimization problems. However, due to the complexities involved in solving optimization problem and due to lack of suitable and efficient solution techniques, they have been transformed and solved as single objective optimization problem. Moreover, because of the presence of conflicting multiple objectives, a multi objective optimization problem results n a number of optimal solutions, known as *Pareto optimal solutions* [29]. In standard practice of using bee's natural properties in computation it has been observed there may be substantial number of instances where selection of best and nearly best solution against very close processes comprising of conflicts or constraints associated with it. This leads to the solution of Pareto type and in this proposal, we introduce the concept of PBCO (Pareto Bee Colony Optimization) in the context of scheduling of several processes very similar to the case of Milk Production Center presented here. Primarily, the standard bee colony and its property have been considered and subsequently their affinity to the process scheduling is discussed.

Algorithm 11.1: Basic Bee Colony Optimization Algorithm-High Level Description.

```
Step 1:    Initialization. Determine the number of Bees and
           the number of iterations I. Select the set of
           Stages ST = {st1, st2, ... , stm}.
           Find any Feasible solution x of the problem.
           This solution is the initial Best Solution.

Step 2:    Forward Pass: Allow Bees to fly from the hive
           and to choose B partial solutions from the set
           of partial solution Sj at stage Stj.

Step 3:    Backward Pass: Send all bees back to the hive.
           Allow bees to exchange information about quality
           of the partial solution created (without
           recruiting nest mates) or dance.

Step 4:    Update Best Solution and return to the nest with
           incrementing the counter.
```

A general scheduling problem can be formalized as follows [30]. We consider a finite set of operations O, partitioned into m subsets $\langle M_1, \ldots, M_2 \rangle =: M(\bigcup M_i = O)$ and into n subsets $\langle J_1, ..., J_n \rangle =: J(\bigcup J_k = 0)$. Together with a partial order $p \subseteq \mathbf{O} \times \mathbf{O}$ such that $p \cap J_i \times J_j$ for $i \neq j$ and a function $p : O \rightarrow N$. A feasible solution is a refined partial order $p* \supseteq p$ for which the restrictions $p * IM_i \times M_i$ and $p * J_k \times J_k$ are total, $\forall i, k$. Most importantly, the cost of a feasible solution is defined by $C_{max}(p/ast) := max\{\sum p(o)|C$ is a chain in $(O, p*)\}$. The effort is to minimize C_{max}. Here M_i is the set of operations that have to be processed on machine i. J_k is the set of operations belong to job k (analogues to core task and sub tasks of the Milk production center mentioned in the case study). All the processes or operations must be performed sequentially and this constraint has been expressed in p^*.

Considering theses assumptions as the benchmark of the case study of the milk production center where the processes are sequential and time bound otherwise the milk core need to be perished out, the model is likely to concentrate on the optimal execution of process scheduling maintaining their intermediate time and other constraints during the makespan. The makespan of the scheduling has been modelled through proposed *Pareto Bee Colony Optimization* (PBCO), envisaging their natural property like waggle dance and foraging.

11.5.1 Waggle Dance –Computational Interpretations

A forager f_i on return to the hive from nectar exploration will attempt with probability p to perform waggle dance on the dance floor with duration D = $d_i A$, where d_i changes with profitability rating while A denotes waggle dance scaling factor. Further, it will also attempt with probability r_i to observe and follow a randomly selected dance. The probability r_i is dynamic and also changes with profitability rating. If a forager chooses to follow a selected dance, it will use the "path" taken by the forager performing the dance to guide its direction for flower patches. We term the path as "preferred path". The path for a forager is a series of landmarks from a source (hive) to a destination (nectar).

11.5.2 Forage and Combining Rough Set

For foraging algorithm, a population of l foragers is defined in the colony. The foragers move along branches from one node to another node in the disjunctive graph and so construct paths representing solutions. A forager must visit every node once and only once in the graph, starting from initial node (i.e. source) and finishing at final node (i.e. sink), so as to construct a complete solution. When a forager is at a specific node, it can only move to next node that is defined in a list of presently allowed nodes, imposed by precedence constraints of operations. It has been observed that after complete deployment and performance benchmark of the proposed BCO uncertainty and ambiguity part still exist. In order to the model more feasible and applicable, the concept of rough set is sorted to use in conjunction with BCO.

11.5.3 Process Scheduling and Optimization under Uncertainty

The scheduling problem has usually been seen as a function of known and reliable information. Modelling approaches developed are mainly deterministic, that is, they are based on nominal or estimated values for all the parameters, thus implicitly assuming that a *predictive schedule* will be executed exactly as planned. However, this assumption is somehow utopian since most plants operate in an unstable and dynamic environment, where unexpected events continually occur. Scheduling problems involve data coming from different sources, and which varies rapidly over time as customer orders, resource availabilities and/or processes undergo changes. Data may be ambiguous, outdated or inaccurately predicted before the problem is solved. Because of the dynamic and uncertain conditions of a real process system, the schedule executed in the plant will probably differ from the predicted one. The effects of the uncertainty may impact on the system's efficiency, eventually leading either to an infeasible situation, or to the generation of opportunities that improve its performance. These situations may become even more significant with the new trends towards managing the whole SC. As stated by Aytug et al. [34], internet technology enables companies within a SC to share their production schedules. In this environment, changes to the production schedule at a downstream node of the SC can cause significant disruptions in upstream operations. These variations can be amplified causing what is known as the *bullwhip effect* [35]. The consideration of the uncertainty when modelling the problem is essential for the development of reliable and effective decision-support systems. Several methodologies are available in PSE (Process Systems Engineering) for optimization under uncertainty. They are categorized, in line with the method used to represent the uncertainty, represented as follows: (a) Probabilistic data - based methods; (b) Stochastic optimization; (c) Fuzzy or Rough Set data - based methods; and (d) Fuzzy Programming.

Blackhurst et al. [36] proposed a network-based methodology to model and analyze the operation of a SC as an abstracted network, with uncertainty in variables such as requirements, capacity, material delivery times, manufacturing times, costs, due dates and priorities. The term stochastic optimization is sometimes used referred to meta-heuristics because of the probabilistic nature of these optimization methods. In general, and as differentiated by some impressive research [37], stochastic optimization involves methods specially developed to address problems with uncertain data, whereas meta-heuristics use stochastic properties in their search.

Although the present model is not deploying stochastic optimization, but broadly the uncertainty part of scheduling events is handled through Rough set based rule metaphor. Hence, the core meta heuristics is being the Pareto Bee colony, subsequently rough set assists to model the associated events with the process and could rank them as well.

11.5.4 Rough Set

Rough set theory is an extension of conventional set theory that supports approximations in decision making. It possesses many features in common (to a certain extent) with the Dempster-Shafer theory of evidence and fuzzy set theory. The rough set itself is the approximation of a vague concept (set) by a pair of precise concepts, called lower and upper approximations, which are a classification of the domain of interest into disjoint categories. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset. For the present problem, the features of individual events have been accumulated:

> A feature x_i is *relevant* if there exists some value of that feature and a predictor output value or A feature x_i is *weakly relevant* if it is not strongly relevant, and there exists some from the set of the features forming a pattern vi, for which there exist subset of features

followed by a ranking of such evaluated features and eventually the choice of the first best m features. Thus ranking of features of all events related to scheduling process of milk food processing industry both on and off the production, including dispatch could be modelled by rough set theory. The content of large-scale data sets containing numerical and categorical information can not be easily interpreted unless the information is transformed into a form that can be understood by human users. The rule extraction algorithms are designed to identify patterns in such data sets and express them as decision rules. The rule extraction concept is illustrated next.

Rule and Data Set

Consider the data set in Table 11.1 with five objects, four features F1-F4, and the decision (outcome).

Table 11.1: Rule Snapshot of Rough Set.

RULE 1 IF (F2 = 0)	THEN	$(D = Low)$; $[2, 6.67\%, 100\%][3, 5]$			
RULE 2 IF (F1 = 0)	AND	$(F4 = High)$	THEN	$(D = 0)$;	$[1, 33.33\%, 100.00\%][1]$
RULE 3 IF (F4 = 0)	THEN	$(D = Medium)$; $[1, 100\%, 100\%][4]$			
RULE 4 IF (F1 = 1)	THEN	$(D = High)$; $[1, 100\%, 100\%][2]$			

The features denote process parameters (e.g., temperature, pressure, time) and the decision is the component performance, high, medium, low. A rule extraction algorithm transforms the data set of Table 11.1 into the decision rules of Table 11.2. The two sets of numbers in square brackets behind each

Table 11.2: Decision rules in Rough Set.

0	0	0	1	Low
0	0	1	3	Low
0	1	0	2	Low
0	1	1	0	Medium
1	1	0	2	High

rule describe its properties. The decision rules of Table 11.1 correspond to the patterns indicated by shaded cells in the matrix in Table 11.2.

11.6 Case Study of Milk Food Product Processing and Production

The raw chilled milk so received subjected to different processes like pasteurization heating and separation/standardization. The detailed process is outlined as follows:

- Raw milk in 40 liters can is received through milk routes established all over the milk shed.
- The milk is graded, weighed an sampled

Problem Statement

The milk production center (MPC) also to undergo different processes and sub processes such as first regeneration, second regeneration, third regeneration, heating, cooling, chilling, internal sub-processing for products, packaging, storage and distribution. The parameters shown in Table 11.3 are crucial for all critical processes and sub processes:

Table 11.3: Process Parameters in Milk Processing Centers.

Process	Sub-Process	Time-Temp-Pressure	Remark
Milk reception	Dispatch	<15 mts	Crucial
Pasteurization	1st Generation, 2nd generation, Heating, Cooling and Chilling	20 sec/ 75-85 o C/25kg/cm2 Stem Pressure	Very Crucial
Standardization	Other Sub processes for Milk based products	5-6 degree Celsius	Important

There are at least 10 major processes identified without split up, which need to be monitored and scheduled priori basis. The processes comprise of different process or makespan of sub processes and its time windows. The conceptual flow diagram of the case study is shown in Fig. fig:MHS-09-1 and its associated parameters are demonstrated in Table 11.3. Among all these parameters certain are influential enough to implicate scheduling of processes, etc.

The broad idea is to incorporate the proposed Pareto Bee Colony Optimization in terms of process scheduling with multi-objectives and constraints. Later on, a few uncertain and ambiguous parameters are separated in scheduling process and rough set approach is coined to address this issue.

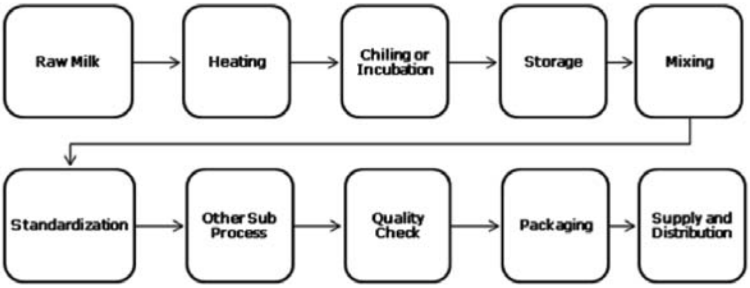

Fig. 11.1: Process Flow Diagram of MPC.

The Mathematical Model

Let us represent the given activity structure as an activity node based network in a milk production center (MPC), whose nodes can be numbered as $1, \ldots, n$. Arcs are used to indicate precedence relationship between nodes (activities). The time required for activity i is denoted by $d(i)$. For given values $d(i)$, the shortest project time δ is defined as the length of critical path. We assume that a finite set $M(i)$ of measures is to each node i. A measure $x_i \in M(i)$ is any means that influences the duration of the process based activity connected with node i. Each measures x_i is realized by certain values:

- The modified duration $d(i, x_i)$ of activity i resulting as a consequence of the measure x_i. The proposed model is concerned in speeding ups the processes and sub processes, so we assume $d(i, x_i) \leq d_i$.
- The cost associated by the process also have been accumulated.
- The model also comprises of target functions like:

$$hf_1(x) = \varphi_1(\delta(x)), \tag{11.3}$$

where, is the shortest project time after reduction of the durations $d(i)$ to the values $d(i, x_i)$ to the values $d(i, x_i)$.

The core process in the milk production center (MPC) is quite exceptional problem in the context of just in time scheduling approach.

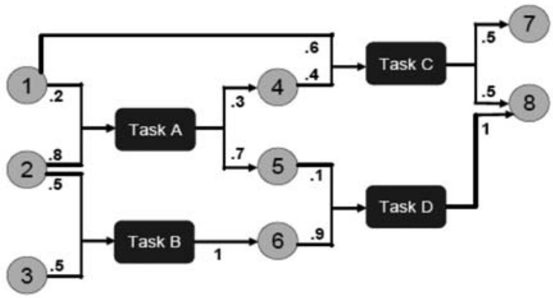

Fig. 11.2: Directed Process Graph of Milk Processing Center.

The present model of milk processing comprising four major tasks or processes and 8 sub processes (Fig. 11.2). The scenario is interpreted through directed graph. Directed graph G, consisting of two sets of nodes, V_1 and V_2, corresponding respectively to materials and tasks. Successor and predecessor nodes to a node in V_1 are always nodes in V_2 and, vice-versa, successor and predecessor nodes to a node in V_2 are always nodes in V_1. Hence the arcs in the graph always connect nodes from V_i to V_j, where $i \neq j$. An arc (r, i) from a node in $r \in V_1$ to a node $i \in V_2$ is introduced if task i requires material r as an input. The label on arc (r, i) is $\rho^{i,r}$, the fraction of input to task i due to material r. Similarly, an arc (i, r) from a node $i \in V_2$ to a node $r \in V_1$ is included in the graph if task i produces material r. The label on arc (i, r) is $\sigma^{i,r}$, the fraction of output from task i in the form of material r. Fig. 11.2 provides an example of such a network with 8 materials (numbered 1-8) and 4 tasks (labelled A-D).

11.6.1 The Proposed PBC Optimization Algorithm

The present problem of Milk Production and its sub process can be mapped with the Pareto Bee colony's characteristics. A forager f_i on return to the hive from nectar exploration will attempt with probability p to perform waggle dance on the dance floor with duration $D = d_i$ A, where d_i changes with profitability rating, while A denotes waggle dance scaling factor. Further, it will also attempt with a probability r_i to observe and follow a randomly selected dance. The probability r_i is dynamic and also changes with profitability rating. If a forager chooses to follow a selected dance it will deploy the path followed by the forager performing the dance to guide its direction for flower patch. In this model we define it as Preferred Process Path (PPP). Hence, the path for a forager is a series of landmarks from a source (hive) to a destination (nectar).

The proposed model of PBCO also implicates a direct relation with the objective function to the profitability rating. Therefore $P_{f_i} = \frac{1}{\varphi_1(\delta(x))}$, where P_{f_i} is the profitability rating for a forager. Theoretically, the average profitability rating of Bee colony is

$$P_{f_{colony}} = 1/n \sum 1/f_1(x) = \varphi_1(\delta((x)),$$

where n is the number of waggle dance just in time, (refer to the process and resources in directed process graph).

Moreover, $1/f_1(x) = \varphi(\delta(x))$ is the value of objective function, which should be maximum if a forage f_i or f_j performs waggle dance. The duration of dance is proportional to the completion of process time, which in turn just in time accomplished between all sub processes. In the process graph, forager must visit each process node i exactly once and it will follow a state transition rule to select best process path, so that no processes of milk production or its sub-process are being delayed. The state transition rule followed by the forager on the process span graph is according to the rule:

$$P_{ij} = \frac{[\rho ij(t)]^{\alpha} \cdot [1/dij]^{\beta}}{\sum [\rho ij(t)]^{\alpha} \cdot [1/dij]^{\beta}}.$$

The rating ρ_{ij} of the directed edge between process nodes i and j is given by:

$$\rho_{ij} = \begin{cases} \alpha \\ 1 - m\alpha/k - m \end{cases}$$

where α is the value assigned to the Preferred Process Path (PPP), $\alpha < 1.0$; k is the number of allowed nodes and m the number of preferred paths. The parameters α and β are the probability of the best process path which is in relation between preferred path versus heuristic distance. According to this rule, edges that are found in the preferred path and that are shorter will have a higher probability to be chosen for the solution. The heuristic distance is the processing time of the operation associated with node j. When a forager completes a full path, the edges it has travelled and the make-span (process span) of the resulting solution will be kept for the waggle dance when it returns to the hive.

Algorithm 11.2: Pareto Bee Colony Optimization -High Level Description

Initialize solution set by the empty set determine the number of bees and iterations
Select the set of processes stage $P_T = \{p_{t1}, p_{t2}, ...p_{tm}\}$
Initilaze bee pheromone matrix τ^k {Store best update solution}
Determine the weight say w_k for each objective k for different processes at random {Start Pareto}
repeat
  for bees 1 to B **do**
    Set $i = 1$
    repeat
      repeat
        Set $j = 1$
      until $j = m$
    until $i = i$
  end for
until termination criteria of each process is satisfied
Call ForwardPass ( ); {Allow bees to fly from the hive and to choose B partial solutions from the set of partial solutions S_j at stage p_{tj}}
for $i = 1$ to n **do**
  select the next process node to traverse according to $P_{ij} = \frac{[\rho_{ij}(t)]^\alpha \times [\frac{1}{d_{ij}}]^\beta}{\sum[[\rho_{ij}(t)]^\alpha \times [\frac{1}{d_{ij}}]^\beta]}$
  $\{Pj \in$ allowed process node$\}$ { where ρ_{ij} is the rating of the edge between node i and j} { d_{ij} denotes heuristic distance between node i and j } { P_{ij} prior probability to traverse from node i to j}
  if the solution x is efficient solution achieved till i^{th} iteration **then**
    update the best solution
    $x := xi$
    Waggle Dance ( ) {profitability rating of Bee colony is

$$Pf_{colony} = 1/n \sum /f1(x) = \varphi 1(\delta((x))$$

    where n is the number of waggle dance through process span}
  end if
  $j = 1$
end for
for each objective k from the solution just found by B bees **do**
  identify ($R - 1$)best solution x^{tk} for object k. $\{\tau^k$ is the bee pheromone matrix for each solution construct x }
end for

11.7 Implementation of PBCO as Multi Objective Optimization

The proposed algorithm has been implemented in C++ on Window XP platform (Pentium IV-2.4 GHz. processor, 256 MB RAM). The milk processing is completely just in time approach as each sub processes are also time bound and material is perishable. The time duration of waggle dance on the process graph is again dependent on the polymorphic method comprising of objective function, shortest process schedule time, and path trace iteration of milk processing. A List type data structure is maintained for and checked against the maximum number of iterations of honeybees across process graph. Practically, the model keeps the traced path in a list, which comprises consecutive operations in pairs. The effective algorithm is involved in operation scheduling and its uncertainty in supply chain process observed in a typical m time bounded milk production unit. The foraging algorithm (waggle dance and nectar exploration) has been incorporated considering bifocal approaches identical with either process or machine centric.

For any process centric approach, a list of currently eligible processes that can be scheduled is always maintained during scheduling process. In order to be viable to process span (makespan), a process's preceding sub-process (of a job) must have been scheduled. Each process planned for Milk production Center in the list is checked against the most recently scheduled sub-processes (on the same machine or parallel processes like cleaning the container, etc.) to identify if the "edge" between the two operations (the most recently scheduled process and the process under consideration from the list) is found in the preferred path. Higher rating ρ_{ij} are assigned to the process with edge found in the path. The scheduling is implemented through Pareto optimal solution, as there are conflict objective among the processes. On the other hand for machine centric approach, a discrete-event simulation and event list of events, which are in sorted order of increasing time, is maintained during scheduling process. At time $t = 0$, events relating to machine-ready status are inserted into the list. Events in the list are removed and executed one by one according to the event time. In case of tie for events having the same time, an event will be randomly picked. For the machine that is associated with the selected event, a list of currently eligible operations will be identified. Each operation in the list is checked against the most recently scheduled operation on the same machine to identify if the "edge" between the two operations is found in the preferred path. Higher rating ρ_{ij} will be assigned to the operation if the edge is found in the path. Although machine centric approach is slightly better than process centric, but the present case study has the variation of processes depending on different milk by products, hence it adopts the process centric approach to test the proposed PBCO algorithm.

11.7.1 Experimental Evaluation

A commercial snippet of data set for a typical Milk Production Center of Asian Country has been accumulated and referred in the case study. The climatic condition is extremely sensitive for Milk and its associated products

The simulation of the data set through C++ coding results some interesting process centric features (see Figs. 11.3 and 11.4. The Pareto distribution of process, optimal time and makespan of processes are presented in the bivariate polynomial form.

Month	Milk procu rement (mill. lit.)	Pasteu rized milk Temp.^{o}C Ideal Val. 75-78	Pasteurized time in seconds Ideal Val. <20	Holding temp ^{o}C Ideal Val. 4-(-1)	Chilling Storage temp. (in Celsius) Ideal Val.5-6	Milk Milk Dis- -patch Time) (in minutes Ideal Val. < 15	Tanker time of Pasteurized milk (in hrs) Ideal Val. > 4.0	Dispatch Milk Products processing time, storage etc. Max. Val. 8-1
DEC. 05	12.6	78	18	4	5	13	3	8
JAN. 06	14	75	20	5	6	12	4.5	9
FEB. 06	12.4	75	19.5	-1	5	13	4	11
MAR. 06	11.4	76	19	-1	5	15	5	8
APR. 06	9	78	19	-1	5	11	4	9
MAY. 06	7. 4	75	17	-1	5	13	4.2	10
JUNE. 06	6.1	77	20	-2	5	14	3.8	10
JULY. 06	5.7	76	20	0	6	12	4	9
AUG. 06	6.6	75	19	3	6	13	4.5	9
SEPT. 06	7.6	77	18	3	5	11	5	10
OCT. 06	9.2	78	17	1	6	12	3	12
NOV. 06	12.6	76	20	1	5	13	4.5	11
DEC. 06	13.8	77	20	4	6	13	4	10

Table 11.4: Dairy data Evaluated through PBCO (Courtesy Jaipur Dairy, Jaipur, India).

The distribution of normal makespan of processes identified in MPC (Milk Production Center) is shown in Fig. 11.3 (Colony Multi-Objective Process Span –Cosine Series Bivariate Order 8) envisaging X, Y and Z axis span. Black Dots are glimpses of bee colony which take care of process span considering source and sink part of process. This denotes the complex multi objective scheduling in the Milk production corresponding the various processes and including time process and temperature. The approach of model is just in time, where the trade off between distribution of bee agents and reinforcement provided by Bee colony is shown by the black dots.

Another distribution map of time (Pareto Optimal Time Span) is shown in Fig. 11.4, scaling and process overlap among the overlapped processes. The peaks of the dots give predictive approach to notify the completion of process in optimal time or either in just in time.

11.7.2 Process Betterment through PBCO - A Comparative Study

To compare and evaluate the performance of the proposed bee colony algorithm, we have included two other meta-heuristics in our experimental study.

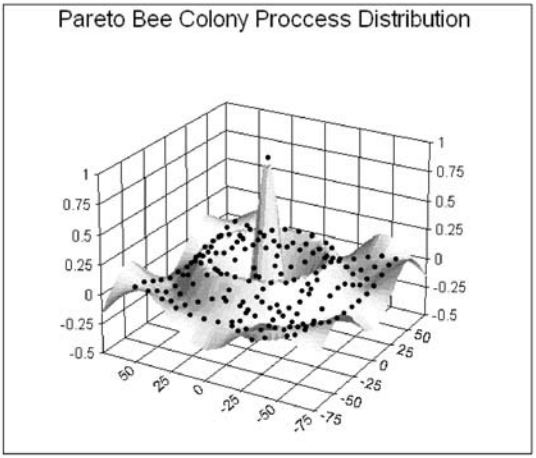

Fig. 11.3: Pareto Bee Colony Process Span, Multi-Objective and Optimal Time Distribution.

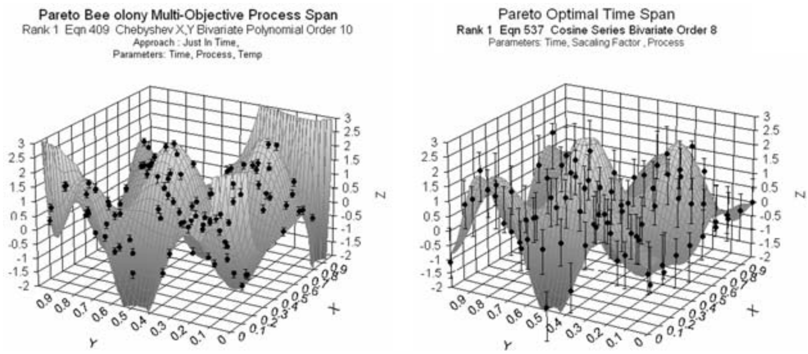

Fig. 11.4: Pareto Bee Colony Process Span, Multi-Objective and Optimal Time Distribution.

The first is an Ant Colony Optimization (ACO) algorithm [32]. The second algorithm is a Tabu Search (TS) algorithm developed by Nowicki and Smutnicki [33].

Within a multi-objective optimization approach, the solutions are compared with respect to their relative “dominance” on both cost and availability objectives. The solutions not dominated by any other are non-dominated solutions. A set (archive) of Pareto-optimal, non-dominated solutions (solutions which are not dominated by any other one) can be collected during the ACO search. Similarly, in order to improve the efficiency of the exploration process, one needs to keep track not only of local information (like the current value of

Table 11.5: Parameters.

Parameters	PBC	ACO	TS
Maximum Number of iterations	1500	1500	1000
Population Size	Process No. (MPC)	Process No No (MPC)	
Weight of Pheromone Trail α	1.0	1.0	
Weight of heuristic value β	2.0	2.0	
Parameter for local updating and profit rating ρ_{ij}	0.8		
Scale Factor	100		
Pheromone Evaporation Coefficient		0.1	
Weight of the availability in the heuristic η	0.3	0.5	
Weight of the cost in the heuristic η	0.3	0.5	
Maximum Number of Elite Solution	25		25
Maximum Size of Tabu List			10

Table 11.6: Performance Comparison of four Meta-heuristics.

Betterment	PBC	BC	AC	TS Process in MPC	Most Critical of Process
Mean percent	10.05	11.2	11.45	6.62	Pasteurization
Maximum	38	38.7	38.6	27.33	Holding and chilling
Most Close Solution schedule without failure	16	15	11	22	For all processes

the objective function) but also of some information related to the exploration process. This systematic use of memory is an essential feature of TS method.

In order to establish comparison, the parameter settings for three algorithms have been presented in Table 11.5.

Table 11.6 elaborates the brief comparison of the process span scheduling in Milk Production Center with three major perspective of meta-heuristics algorithms. Although results exhibit that TS is the smartest choice among the four, it provides most likelihood results within smallest execution time. Practically, TS has the advantage to solicit the best solution and it takes care to the Tabu list, instead of constructing the solution from source to sink applied to Bee and Ant Colony. The proposed Pareto Bee is slightly better than standard Bee Colony, where as broadly different from ACO heuristics in the context of process scheduling. The inclusion of rough set is a primary back up for the associated events with the process and could rank them as well under diversified condition.

11.8 Conclusions and Future Work

A hybrid meta-heuristic approach based on a multi-objective Bee Colony algorithm combined with constructive rough set heuristics is a valuable decision support tool for planning operations in a supply network for rapidly perishable material for food processing industry. Provided a detailed mathematical model of the supply network, our experimental investigation shows that such a hybrid approach is able to provide an effective scheduling algorithm. This work also provides a comparative platform on bio-inspired algorithms and rough set (e.g. Bee Colony, Ant Colony and Tau Search) used for scheduling in food and beverage processing industry. The involvement of multi-objective process span through Pareto scheduling also is proposed in this chapter. Further hybridization with other soft computing approach like fuzzy logic [31] could be developed on honey bee algorithm. The more extension on rough set in process scheduling is also expected. In this work, exact preferred path of the process schedule span has not been evaluated; rather we incorporated local search heuristics. This is because the processes and sub processes in the case study of Milk production Center are somewhat static in nature. It has been observed that among the all bio-inspired and stigmergic formulation of algorithm, ant colony is the most prominent one, but there are different agent based application areas where the agents work in a completely distributed environment and thus maintaining the pheromone transition table becomes slightly impractical. In those cases, Bee colony could be a better alternative.

Acknowledgement

The present case study and data has been considered from a milk food product processing *Jaipur Dairy* and its production house based on Jaipur, India. The work is partially supported by Department of Mechanical Engineering, Malviya National Institute of Technology, Jaipur, India.

References

1. Box, G.E.P., Hunter, W.G. & Hunter, J.S. (1978) Statistics for experimenters: an introduction to design, data analysis, and model building, Wiley, New York.
2. Myers, R.H. & Montgomery, D.C. (2002) Response surface methodology: process and product optimization using designed experiments, Wiley, New York
3. den Besten, L.M. Simple Meta-heuristics for Scheduling. An empirical investigation into the application of iterated local search to deterministic scheduling problems with tardiness penalties, PhD Thesis, October 2004.
4. Nareyek, A. (2000): AI Planning in a Constraint Programming Framework, in Proceedings of the Third International Workshop on Communication-Based Systems (CBS-2000)

5. Kreipl, S., Pinedo, M. Planning and Scheduling in Supply Chains: An Overview of Issues in Practice, Production and Operations Management Vol. 13, No. 1, spring 2004, pp. 77-92.
6. Chopra, S. and Meindl, P. Supply Chain Management: Strategy, Planning, and Operations: Prentice Hall College, 2001
7. Harrison, T.P., Lee, H.L. and Neale, J.J. The Practice of Supply ChainManagement: Kluwer Academic Publishing, 2003.
8. Truong T.H and Azadivar, F. Simulation based optimization for supply chain configuration design, presented at the Winter Simulation Conference, Piscataway, NJ, 2003.
9. Joines, J.A., Kupta, D., Gokce, M.A., King, R.E. Supply Chain Multi-Objective Simulation Optimization, presented at the 2002 Winter Simulation Conference, 2002.
10. Al-Mutawah, K., Lee, V., Cheung, Y. Modeling Supply Chain Complexity using a Distributed Multi-objective Genetic Algorithm, presented at The 2006 International Conference on Computational Science and its Applications ICCSA'06, Glasgow, Scotland, 2006.
11. Simchi-Levi D., Kaminsky, P. and Simchi-Levi, E. Designing and Managing the Supply Chain, McGraw-Hill, 2000.
12. Ribeiro, R. and Lourenço, H.R. A multi-objective model for a multi-period distribution management problem Economic Working papers Series, Universitat Pompeu Fabra, Spain, 2001.
13. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Articial Systems. Oxford University Press (1999).
14. Sumpter, D., Pratt, S. A modelling framework for understanding social insect foraging. Behavioral Ecology and Sociobiology 53 (2003) 131-144.
15. Goss, S., Deneubourg, J.L. The self-organising clock pattern of Messor pergandei (Formicidae, Myrmicinae). Insectes Soc. 36:339-346, 1989.
16. Bartholdi, J.J., Seeley, T.D., Tovey, C.A., Vande Vate J.H. The pattern and effectiveness of forager allocation among flower patches by honey bee colonies. J. Theor. Biol. 160:23-40, 1993.
17. Beckers, R., Deneubourg, J.L., Goss, S., Pasteels, J. Collective decision making through food recruitment. Insectes Soc. 37:258-267, 1990.
18. Beckers, R., Deneubourg, J.L., Goss, S. Modulation of trail laying in the ant Lasius niger (Hymenoptera: Formicidae) and its role in the collective selection of a food source. J Insect Behav 6:751-759, 1993.
19. Beekman, M., Sumpter, D.J.T., Ratnieks, F.L.W. Phase transition between disordered and ordered foraging in Pharoah's ants. Proc. Natl. Acad. Sci. USA 98:9703-9706, 2001.
20. Bonabeau, E. Comment on Phase transitions in instigated collective decision making. Adapt. Behav. 5:99-105, 1997.
21. Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M. The self-organizing exploratory pattern of the Argentine ant. J Insect Behav 3:159-168, 1990.
22. Goss, S., Deneubourg, J.L. The self-organizing clock pattern of Messor pergandei (Formicidae, Myrmicinae). Insectes Soc 36:339-346, 1989.
23. Nicolis, S.C., Deneubourg, J.L. Emerging patterns and food recruitment in ants: an analytical study. J. Theor. Biol. 198: 575-592, 1999.
24. Biesmeijer, J., de Vries, H. Exploration and exploitation of food sources by social insect colonies: a revision of the scout recruit concept. Behav. Ecol. Sociobiol 49:89-99, 2001.

25. Traniello, J.F.A. Recruitment behavior, orientation, and the organization of foraging in the carpenter ant Camponotus pennsylvanicus DeGeer (Hymenoptera: Formicidae). Behav Ecol Sociobiol 2:61-79, 1997.
26. Holdobler, B., Stanton, R.C., Markl, H. Recruitment and food retrieving behavior in Novomessor (Formicidae, Hymenoptera), I. Chemical signals. Behav. Ecol. Sociobiol 4:163-181, 1978.
27. Reuter, M., Keller, L. Sex ratio conflict and worker production. Am. Nat. 158:166-177, 2001.
28. Hall, R.W. Driving the Productivity Machine: Production Planning and Control in Japan. American Production and Inventory Control Society, Falls Church, Vancouver, 1981.
29. Deb, K. Multiobjective Evolutionary Algorithms: Introducing Bias among Pareto-Optimal Solutions. In A.Ghosh and S. Tsutsui(Eds.), Theory and Applications of Evolutionary Computation: Recent Trends, Spinger -Verlag, London, 2002.
30. Blum, C. ACO applied to Group Shop Scheduling: A case study on Intensification and Diversification. In Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS 2002), 2002. Also available as technical report TR/IRIDIA/2002-08, IRIDIA, Universite Libre de Bruxelles.
31. Teodorvic, D., Dell'orco, M. Bee Colony optimization- A Cooperative Learning approach to Complex Transporation Problem, Advanced OR and AI Methods in Transportation, 2005, pp 51-60.
32. Dorigo, M., Maniezzo, Vittorio, Colorni, Alberto, Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics,Part B: Cybernetics, 1996. 26(1): p. 29-41.
33. Nowicki, E. and Smutnicki, C., A fast taboo search algorithm for the job shop problem, Management Science, Vol. 42, No. 6 (1996), pp. 797-813.
34. Aytug, H., Lawley, M. A., McKay, K., Mohan, S., Uzsoy, R. M., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Research 161, pp. 86-110.
35. Lee, H. L., Padmanabhan, V., Whang, S., 1997. The bullwhip effect in supply chains. Sloan Management Review 38, pp.93-102.
36. Blackhurst, J., Wu, T., O'Grady, P. Network-based approach to modeling uncertainty in a supply chain. International Journal of Production Research 42 (8), pp.1639-1658, 2004.
37. Fu, M. C. Simulation optimization. In: Peters, B. A., Smith, J. S., Medeiros, D. J., Rohrer, M. W. (Eds.), Proceedings of the 2001 Winter Simulation Conference.