ANTS SYSTEMS, A STATE OF THE ART OVERVIEW: APPLICATIONS TO INDUSTRIAL PLANTS PROBLEMS

Francesco Longo, Giovanni Mirabelli, Sabrina Rondinelli
Modeling & Simulation Center, Laboratory of Enterprise Solutions (MSC-LES)
Mechanical Department, University of Calabria
Via P. Bucci, Cubo 44C, 87036 Rende, Italy
f.longo@unical.it, g.mirabelli@unical.it, srondinelli@unical.it

KEYWORDS
Ants Systems, job shop systems, Simulation

ABSTRACT

This paper presents a state of the art overview on Ants Systems. Initially the work proposes the fundamental concepts of Ants Theory and offers a description of the main algorithms (based on Ants Theory) and the respective fields of application. Among different fields of application, the paper focuses on industrial plants scheduling problems in open shop, flow shop and job shop systems. The last part of paper discusses about the possibility to combine Modeling & Simulation and Ant Theory to develop advanced real-time support tools for the job shop scheduling problem.

INTRODUCTION

The study of ants' behavior has always interested numerous scientists all over the world. The ability of these social insects to organize and cooperate each other arouses particular interest and curiosity. In particular, Ant's Theory takes inspiration from the observation of ants' behavior when they search for food. In fact, ants are capable of finding the shortest path from source food to their nest and vice-versa, but above all ants are capable of exchanging such information using a simple form of indirect communication (stigmergy). This is possible since ants, while walking from food source to nest, deposit on the ground a particular substance called pheromone, forming a pheromone trail that the other ants can smell. When ants choose the way to follow, they prefer, in probability, paths marked by strong pheromone concentration. In order to better understand ants' behavior we report a n example taken from Dorigo et al. (1996).

In figure 1 ants follow a path between the nest (position in A) and the food (position in E). Suppose that, at certain point in time, an obstacle appears between points A and E. Then, ants have to decide the path to follow: AHE or ACE. As the Figure 2 shows that the two paths are not equal: route AHE is shorter than ACE. Initially ants choose the way to go random: on the average, middle of all ants goes through path AHE and the other half goes through path ACE. Subsequently the choice is influenced by the intensity of pheromone trails that ants smell on the path.

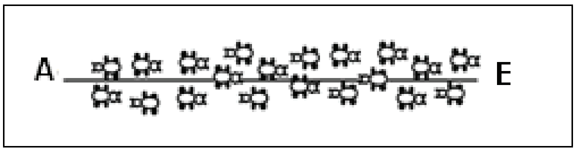

Figure 1: Ants follow a path between points A and E. (Dorigo et al. 1996a)

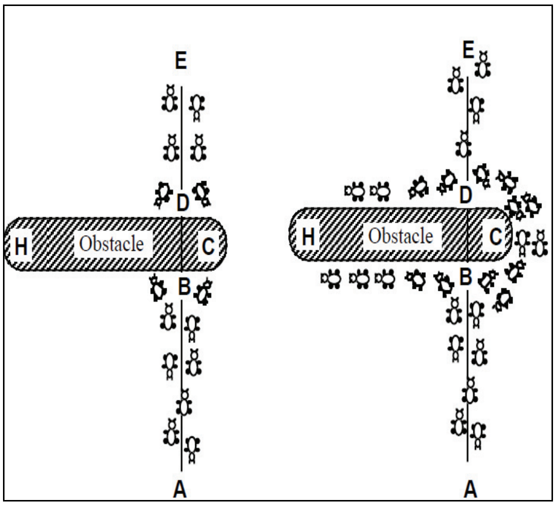

Figure 2: An obstacle is interposed and ants can choose to go around it follow one of the two different paths. (Dorigo et al. 1996a)

Ants which choose path ACE spend less time than ants choosing the other path, than the smell of pheromone trails on the first path is more intense than on the longer one (in fact with passing of time pheromone evaporates). Then, ants that return from E to A will find a stronger trail on path ECA and will choose this way. This process continues time after time until all the ants colony will go on the shorter path: ants have been able to identify the optimum tour. It is interesting to note that each ant gives a share for finding the optimal solution, but it is only the ants' colony which finds the shorter path. This is possible using a simple form of indirect communication mediated by pheromone, known as stigmergy (Grassé 1959).

The first author that developed an algorithm based on ants' behavior was Marco Dorigo (Dorigo 1992; Dorigo 1993), followed, after some time, from Luca Maria Gambardella (Gambardella et al. 1997; Gambardella

Proceedings 23rd European Conference on Modelling and Simulation ©ECMS Javier Otamendi, Andrzej Bargiela, José Luis Montes, Luis Miguel Doncel Pedrera (Editors)
ISBN: 978-0-9553018-8-9 / ISBN: 978-0-9553018-9-6 (CD)

and Dorigo 1995) who proposed several algorithms based on the same principles. While Ant's Theory was spreading, many authors have focused on this research topic, developing algorithms for different fields of application.
Before getting into details of the state of the art overview, in the sequel a brief description of the paper sections is reported. Section 2 proposes a brief description of the main algorithms developed and the respective fields of application. In section 3 the attention is focused on algorithms applied to industrial plants problems (with a description of different cases). Section 4 discusses about one of the most important problems in industrial plants: job-shop scheduling problem. Finally, the last section summarizes conclusions and research activities still on going.

ANT'S THEORY ALGORITHMS

The first model based on Ant's Theory was proposed by Dorigo, it was a new heuristic approach for the solution of combinatorial optimization problems called Ant System (Dorigo et al. 1991; Dorigo 1992; Dorigo et al. 1996). Ant System was first applied to solve traveling salesman problem. Over the years several versions of AS have been developed to improve and extend the performances of the initial algorithm and apply it to solve other optimization problems. In order to develop a common framework for all the versions of the AS, in 1999 was proposed the Ant Colony Optimization (ACO) meta-heuristic (Dorigo and Di Caro 1999a).
Today there are numerous successful implementations of the ACO meta-heuristic applied to different combinatorial optimization problems. It is possible to distinguish two different classes of problems: static combinatorial optimization problems and dynamic combinatorial optimization problems. Static problems are those in which the characteristics of the problem not change while the problem is being solved. On the other hand, dynamic problems are those in which the characteristics of the problem evolve during its resolution. In the following is reported a brief description of the main problems solve with ACO algorithms.

Traveling salesman problem

The first problem solved by an algorithm based on Ant's Theory was the traveling salesman problem (TSP). TSP consists of finding the shortest route visiting each member of a collection of locations and returning to the starting point. At the beginning the algorithm used to solve TSP was the AS, in which the ants build optimum solutions (finding feasible tours of minimum costs) by moving on the problem graph from city to city. Considering the way to update the pheromone trail, three different versions of the AS have been defined: ant-density, ant-quantity and ant-cycle. In the first two versions ants deposit pheromone while building a solution, instead in third version they deposit pheromone only when a complete tour has been built.
The MMAS algorithm (Max-Min AS, Stützle and Hoos 1997a; Stützle and Hoos 1997b) is similar to the AS but, in this version, the pheromone trail is updated offline. The way to update the pheromone trail is the same proposed by Dorigo and Di Caro (1999b): "…the arcs which were used by the best ant in the current iteration receive additional pheromone". The pheromone trail values are restricted at the interval $[\tau_{min},\tau_{max}]$, and trails are initialized at maximum value τ_{max}.
Another version is the AS_{rank} (Bullnheimer et al. 1997) in which, as in the MMAS version, the arcs that have been visited by ants that carried out the best tour are updates with an additional quantity of pheromone (this is equivalent to AS's elitist ants pheromone updating) The difference is that the ants are ranked by tour length they complete.
The Ant-Q (Dorigo and Gambardella 1995; Gambardella and Dorigo 1996) is an algorithm that tried to fuse the AS and the Q-learning (Watkins 1989). The idea of this algorithm is to update step-by-step pheromone trails with a quantity which is a prediction of the value of the next state. Suddenly it appears that the evaluation of this predictable value was extremely difficult, than Ant-Q was abandoned and replaced by ACS (Ant Colony System; Dorigo and Gambardella 1997a; Dorigo and Gambardella 1997b). The ACS gives approximately the same performances of Ant-Q but it is simpler to use. The two algorithms simply differ in the value used to update pheromone trails.
The ACS is a further version proposed to improve the performance of the AS, that gives good solutions in reasonable time only for small problems. The rationale of ACS and AS is the same but some differences exist:

- In ACS the global pheromone trails are updated offline: at the end of an iteration a quantity of pheromone is added to the arcs used by the ant that complete the best tour;
- Ants use a different decision rule, called pseudo-random-proportional rule, to choose the next city to visit;
- Ants update the pheromone trails only on-line, step-by-step, during the construction of a solution;
- ACS algorithm uses a data structure, called candidate list, that allows to have additional local heuristic information.

Quite similar to ACS is another algorithm: ACS-op-3 (Freisleben and Merz 1996a; Freisleben and Merz 1996b). The difference is that the ACS-op-3 activates a local search procedure (based on variant of the 3-opt local search procedure) to improve the solutions generated by ants.

Quadratic Assignment Problem

QAP is the problem to assign n resources to n locations trying to minimize total allocation cost (this cost is a function of the way to allocation chosen). This kind of problem is a generalization of TSP. As a consequence the algorithms used to solve a QAP are generalization of algorithms used to solve the TSP. Further information can be found in Maniezzo et al. (1994) that propose the AS-QAP algorithm and in Stützle and Hoos (1998) that propose the MMAS-QAP algorithm.

Job scheduling problem

The problem is the following: given M machines and a set of J jobs it is necessary to choose the order sequence of operations to be executed on these machines so that the maximum of the completion times of all operations is minimized and no two jobs are processed at the same time on the same machine. The algorithm used to solve this problem is the AS, but since exist some differences in nature of the constraints of this problem respect to the TSP, it is necessary to define a new way of building the ants' tabu list. Currently AS-JSP algorithm (Colorni et al. 1994) is used to solve problems of maximum dimensions up to 15 machines and 15 jobs and the generate results are not satisfactory. This results suggest that this field of applications should be further investigated.

Vehicle Routing

Vehicle Routing problem is the following: consider a graph G=(N,A,d), where N is the set of nodes, A is the set of arcs and d_{ij} is a weight associated to arc (i,j) that represent the distance between node i and j. Node 0 represents a deposit in which are localized vehicles (each one of capacity C), while the other nodes represent customers that should be served and ask a quantity D at time T. The problem is to find the cycle of minimum cost such that every customer is visited and its demand dispatched respecting vehicle capacity constrains and assuring that every vehicle starts and ends its tour in the depot. The first algorithm proposed for this problem is AS-VRP that is an extension of the AS based on AS_{rank}. Another algorithm for the Vehicle Routing problem is HAS-VRP (Gambardella et al. 1999). This algorithm first reformulates the initial problem by adding to the city a set of M-1 depots, where M is the numbers of vehicle. In this way the initial problem becomes a TSP with additional constraints. Then it is applied HAS-VRP algorithm, which is inspired by ACS. Specific research works in the field of vehicle routing have also been developed by De Sensi et al. (2006) and Curcio et al. (2007). The first research work presents an AS algorithm integrated in a simulation model for supply chain vehicle routing optimization. The second one compares the AS and Genetic Algorithms (GA) for vehicles routing optimization in a pharmaceutical distribution system.

Sequential Ordering

The sequential ordering problem consists in finding minimum weight Hamiltonian cycle on a directed graph with nodes and arcs weighted and trying to respect precedence constraints among nodes. In this case the proposed algorithm is HAS-SOP (Gambardella e Dorigo 1997), an extension of ACS. In fact, in HAS-SOP, unlike ACS, the set of feasible nodes is built taking in consideration the precedence constraints and, for local search, it is use a variant of 3-opt procedure.

Graph Colouring

Graph Colouring is the problem to find minimum number of colours for colouring a graph. The algorithm proposed is ANTCOL (Costa and Hertz 1997) that makes use of well-known graph colouring heuristics *recursive large first* (RLF) (Leighton 1979) and DSATUR (Brelaz 1979).

Shortest Common Supersequence

The shortest common supersequence (scs) is a common supersequence of minimal length given two sequences $A=<a_1,\dots,a_m>$, $B=<b_1,\dots,b_n>$. In the shortest common supersequence problem, the two sequences **A** and **B** are given and the objective is to find the shortest possible common supersequence of these sequences. In general, the scs is not unique. This problem is solved by an algorithm called AS-SCS (Michel and Middendorf 1998), which differs from AS classic because AS-SCS uses a look-ahead function which takes into account the influence of the choice of the next symbol to append at the next iteration.

Algorithms for dynamic combinatorial optimization problems

Table 1 lists all the algorithms presented in the previous sections also reporting authors and field of application.
The algorithms for dynamic combinatorial optimization problems have focused on communications networks. In fact, this kind of problem is characterized by dynamic conditions. ACO implementations for communications networks are grouped in two classes: connection-oriented and connection-less networks. In the first case all the packets follow a common path in a preliminary phase of selection, instead in the second one all packets can follow different paths. Algorithms proposed for the two classes are reported in the next table (Table 2) with a list of references.

Table 1: List of applications of ACO algorithms to static combinatorial optimization problems (Dorigo and Di Caro 1999).

Problem Name	Authors	Algorithm Name
Traveling Salesman	• Dorigo, Maniezzo & Colorni, 1991 • Gambardella & Dorigo, 1995 • Dorigo & Gambardella, 1996 • Stützle & Hoos, 1997 • Bullnheimer, Hartl & Strauss, 1997	• AS • Ant-Q • ACS & ACS-3-opt • MMAS • AS_{rank}
Quadratic Assignment	• Maniezzo, Colorni & Dorigo, 1994 • Gambardella, Taillard & Dorigo, 1997 • Stützle & Hoos, 1998 • Maniezzo & Colorni, 1998 • Maniezzo, 1998	• AS-QAP • HAS-QAP • MMAS-QAP • AS-QAP • ANTS-QAP
Job-shop Scheduling	• Colorni, Dorigo & Maniezzo, 1994	• AS-JSP
Vehicle Routing	• Bullnheimer, Hartl & Strauss, 1996 • Gambardella, Taillard & Agazzi, 1999 • De Sensi et al., 2006 • Curcio et al., 2007	• AS-VRP • HAS-VRP • AS • AS & GA
Sequential Ordering	• Gambardella & Dorigo, 1997	• HAS-SOP
Graph Coloring	• Costa & Hertz, 1997	• ANTCOL
Shortest Common Supersequence	• Michel & Middendorf, 1998	• AS-SCS

Table 2: List of applications of ACO algorithms to dynamic combinatorial optimization problems (Dorigo and Di Carlo 1999)

Problem Name	Authors	Algorithm Name
Connection-oriented network routing	• Schoonderwoerd, Holland, Bruten & Rothkrantz, 1996 • White, Pagurek & Oppacher, 1998 • Di Carlo & Dorigo, 1998 • Bonabeau, Henaux, Guérin, Snyers, Kuntz & Théraulaz, 1998	• ABC • ASGA • AntNet-FS • ABC-smartants
Connection-less network routing	• Di Carlo & Dorigo, 1997 • Subramanian, Druschel & Chen, 1997 • Heusse, Guérin, Snyers & Kuntz, 1998 • Van der Put & Rothkrantz, 1998	• AntNet & AntNet-FA • Regular ants • CAF • ABC backward

The next section focuses on ants' algorithms developed for specific applications in industrial field.

ANT'S THEORY APPLIED TO PRODUCTION SCHEDULINGS IN INDUSTRIAL PLANTS

The problem of production scheduling within industrial plants consists of scheduling jobs (i.e. shop orders) on the available machines over the time. The production scheduling problem allows multilevel classification:

1) Depending on the number of machines:
   - Scheduling on single machine: in this case it is necessary to find the sequence of jobs that minimize use time of a single machine (in literature known as "problems of pure scheduling);
   - Scheduling on multi-machines: in this case jobs should be processed on more machines.
2) The scheduling on multi-machines can be also divided in:
   - Scheduling on homogeneous machines: all machines execute the same operation and jobs can be processed indifferently on each machine. The problem is twofold: assignment of each job to one of the alternative machines (first level: assignment problem); jobs scheduling on each machine (second level: scheduling problem), with the aim to optimising an objective function.
   - Scheduling on heterogeneous machines: each machine executes a particular operation.
3) For jobs scheduling the following case have to be considered:
   - Open shop: jobs do not have a fixed machines visiting order;
   - Flow shop: jobs have to be processed through a fixed sequence of machines and the processing operation order of each job is the same.
   - Job shop: jobs have to be processed through a sequence of machines, but the processing operation order as well as the sequence of machines of each job is different from other.

The following figure 3 represents schematically the previous scheduling problems classification within industrial plants. In the following we focus the attention first on single machine problem and then on the last three problems (open, flow and job shop) and, in particular, we present the ACO algorithms proposed to solve these problems.

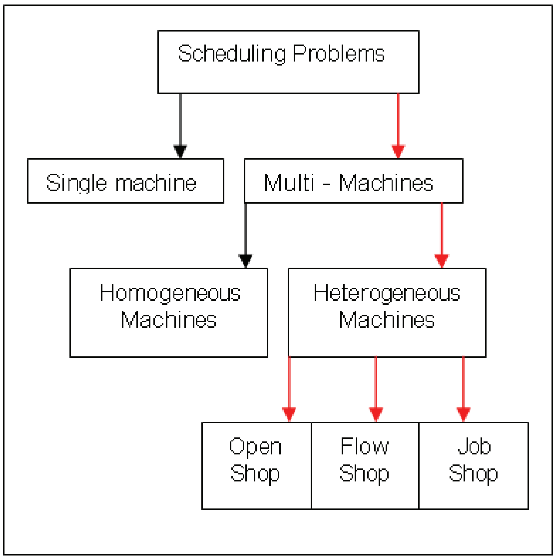

Figure 3: Classification of production scheduling problems.

Single Machine

In this case jobs utilize the same resource (machine). In a real world system (i.e. an industrial plant) several elements make this problem very complex: process time, due date, and above all setup time. In literature, we find that this problem has been classified as NP-hard and its description is as follows: there are n jobs to be processed on the same machine; the machine can process only one job at a time. Each job is characterized by: processing time, due date, weight in terms of cost for time unit and setup time. To solve this problem an ACO heuristic has been presented by Liao & Juan (Liao & Juan 2005). The ACO algorithm proposed by Liao & Juan is basically the ACS version but it introduces the minimum pheromone value from "max-min-ant-system" (MMAS). The algorithm has two additional distinctive features: it introduces a new parameter for the initial pheromone trail and adjusts the timing of applying local search.

Another ACO algorithm for the same problem was proposed by Gravel (Gravel et al. 2001). In this case the algorithm is developed on a real industrial case study: a scheduling problem in an alcan aluminum foundry. In this algorithm the ACO meta-heuristic allows to perform a multi-objective optimization: minimization of unused capacity, minimization of the total tardiness of the set of orders, minimization of the total number of draining required when changing materials, minimization of transportation function penalty that computes the total unused vehicle capacity. To adapt the algorithm the authors implement a distance matrix D that contains information derived from each of the objectives and, on the other hand, a new rule for updating the pheromone trail at the end of a cycle based on one of the objectives.

Open Shop

The Open Shop problem can be defined as follows: a finite set of jobs, O, that have to be processed on a set of machines, M. Jobs do not have a fixed order to visit machines and each machine can process one job at a time. The objective is to minimize the makespan (the maximum of jobs completion time).

The algorithm proposed to solve this problem, based on Ant's Theory, is Beam-ACO (Blum, 2005). This algorithm is an union between Ant Colony Optimization and Beam Search (Ow PS 1988), a classical tree search method that is a derivation of branch and bound algorithm. Beam-ACO works as follows: for each iteration, *n* ants perform a probabilistic beam search and each ant constructs up to k solutions. Then, depending on a measure, called convergence factor, and a boolean control variable, an update of the pheromone values is performed. After, the convergence factor is recomputed and it is decided if the algorithm has to be restarted. The algorithm stops when the termination conditions are satisfied.

Flow Shop

The simplest case of Flow Shop is the 2-machine flowshop scheduling problem with *n* jobs to schedule on 2 machines. Every jobs have a specific processing time on the two machines (i.e. job *i* has processing time a_i on first machine and b_i on second machine). C_i is the completion time of job *i* on machine 2, where i=1,…n. For these types of problems two criterions can be defined: (i) the makespan criterion, called Cmax, defined as the maximum completion time of jobs on machine 2; (ii) the total completion time criterion, called $\sum$Ci, defined as the sum of completion time of jobs on machine 2. It is supposed that the total completion time criterion has to be minimized subject to the condition that the makespan computed is minimum. It is supposed to minimize the makespan (maximum completion time) as the primary objective and minimize total flow-time (sum of the completion times of all jobs) as the secondary objective. In practice this problem is a multiple criteria scheduling problem and it is a strongly NP-hard.

The 2-machine flowshop scheduling problem has been extensively studied in literature and both exact and heuristic algorithms have been proposed. An ACO algorithm was also proposed to solve this problem.

The ACO heuristic proposed for this problem is known as SACO (T'kindt et al. 2001). In SACO each ant builds a feasible schedule by using a constructive procedure. This procedure uses a particular structure called "pheromone matrix" that, based on memory shared by ants, contains the information about the ways to build a good feasible schedule. Every element that composes the pheromone matrix represents a probability (i.e. τ_{ij} is the probability of having job i at position j in a good schedule for the $\sum$Ci criterion). Starting from position 1 to position n, the most suitable job for position j is chosen according to either the intensification mode (the ant chooses the job with the highest value of τ_{ij}) or the diversification mode (the ant uses a wheel process to select the most suitable job). When an ant has built a complete schedule, a local search is applied, that among various schedules keeps the schedule that has an optimal value of the makespan and the lowest value for the total completion time. Several computational experiments have suggested that SACO yields better results than existing heuristics.

Another variant of 2-machine flowshop scheduling problem involve the machine setup time. In this case the objective function is to find a schedule of jobs such that the total completion time, the sum of completion times of all jobs, is minimized. To solve this problem was proposed a particular variant of ACO (Shyu et al. 2004). First of all the original problem is transformed into a graph-based-model in which every edge has an associated weight *w*. Also, unlike the strategy used by Dorigo in classical ACO, where every edge has a constant quantity of initial pheromone, this method initializes the pheromone on edges on the basis of results obtained by a greedy heuristic algorithm. To obtain a better initialization of pheromone intensity, the greedy heuristic is applied *n* times by letting every node be the starting node. Obtained the initial solution the algorithm starts choosing the next node to visit and updating pheromone trail. The algorithm ends after a given number of iterative cycles.

The flow shop problem becomes more complicate in case of *n* machines. Rajendran & Ziegler (2002) propose two algorithms to solve this problem. The first algorithm, called M-MMAS, extends the ant-colony algorithm called MMAS, by incorporating the summation rule developed by Merkle and Middendorf (2000) for the single-machine total weighted-tardiness problem, and modifying the procedure for job selection to be appended to the partial ant-sequence. In addition, the M-MMAS makes use of a new local search technique, called job-index-based local search. The second algorithm developed by Rajendran & Ziegler (2002) is called PACO. The first sequence of the ant-colony algorithm is obtained in a way similar to M-MMAS with the objective function value set to best value. There is a specific procedure to select an unscheduled job *i* for position *k*. The resulting sequence is subjected to the job index-based local search scheme three times to improve the solution. Moreover the algorithm has a procedure to update pheromone trail based, not only on the resultant sequence obtained after the three-time application of the job index-based local search procedure, but also on the relative distance between a given position and the position of job *i* in the resultant sequence. PACO algorithm generates 40 ant-sequences and among them it is obtained the best heuristic sequence.

Another important contribution is the application of ACS to solve flow shop problem, in which n jobs should be processed on m machines, with the objective to find a permutation of jobs that minimizes the makespan. The problem can be represented in ACS by a disjunctive graph, where the set O contains all operations of each job on machine, C contains the precedence relationships between the processing operations of a single job, and D represents the machine constraints of operations belonging to different jobs. Also, there are two additional nodes: a nest node N and a source food node F. After the construction of the graph, since the jobs have all the same ordering sequence, it is simple to find the first sequence. The next node an ant chooses to move to is calculated by applying a specific state transition rule. The chosen node is then added to the tabu list and the process is iterated. At the end, the node permutation given by the tabu list can determine the job sequence.

Another algorithm to solve the flow shop scheduling problem was proposed by Gajpal et al. (2005), called NACO. This time the aim is to minimize the variance of jobs completion times (called the completion-time-variance problem, CTV). In the NACO algorithm the initial solution is obtained by NEH (Nawaz, Enscore and Ham) heuristic considering the jobs CTV minimization. To improve the initial sequence is used a random-job-insertion local search procedure and for every ant-sequence generated and for final solution is executed the local search procedure three times. As in every ACO algorithm, there is a step to update the pheromone trail or trail intensity.

Job Shop

Among different scheduling problems previously presented the job shop is the most difficult and problematic to handle. Nowadays most of the industries, developing FMS (Flexible Manufacturing Systems), have to deal with job shop scheduling problem. As a consequence numerous algorithms and procedures have been developed over the time to handle this problem. In the last years also algorithms based on ACO meta-heuristic have been proposed.

The first algorithm handles the problem of FJP (flexible job shop scheduling) which extends the classic problem of job-shop scheduling. In fact, in a very large number of industries all available machines are divided into groups of identical machine tools. This version of job shop scheduling problem is known as FJS with parallel machines. A scheduling algorithm was introduced by Rossi and Dini (2006) and represents the FJS problem with separable transportation and sequence dependent setup times in a disjunctive graph. The aim of the algorithm is to minimize the makespan. The algorithm proposed is ACS with a number of innovative skills: the first is an approach based on the disjunctive graph model and a LS (List Scheduler) algorithm (that drives an ant to visit the digraph model). The second feature is the pheromone trail structure that is based on the before mentioned disjunctive graph model. Finally, two original components, the routing-precedence-based visibility function and the method to approximate non-delay schedules are introduced to improve the performance.

For FJS problem another important procedure have been obtained combining GA (genetic algorithm) with ACO (Rossi & Boschi 2008) as a mechanism of collaborative search. This procedure presents new skills in the methods of selecting the GA subpopulations and updating pheromone trails. This algorithm has been implemented in a real system to test its performances; the results show that the algorithm is a powerful mechanism for enhancing the searching capability of both genetic algorithms and ant colony systems.

ACO AND MODELLING AND SIMULATION

This paragraph discusses about simulation use in job shop scheduling problem (in real industrial environments) and in particular about the implementation of simulation models supported by ACO meta-heuristic.

The job shop scheduling problem within real industrial environments is usually characterized by a number of restrictive assumptions (i.e. complete resources utilization, no preemption, a job cannot simultaneously occupy more than one machine at a time, etc.). Restrictive assumptions generate ideal scenarios, useful

to gain confidence and knowledge about the real system, but they do not give the possibility to transfer and apply final results to the real system. Usually restrictive assumptions can be deleted using Modeling & Simulation to recreate systems complexity.
In the following we report an example of simulation model to solve the flexible job shop scheduling problem. This work, proposed by Li-Ning Xing et al. (2008), considers the following restrictive assumptions for the job shop scheduling problem:

- Jobs are independent from each other;
- Machines are independent from each other;
- Setting up time of machines is negligible;
- Move time between operations is negligible;
- At a given time, a machine can execute at most one operation;
- No more than one operation of the same job can be executed at a time;
- There are no precedence constraints among the operations of different jobs.

The aim of a simulation model is to assign the operations to machines and to schedule the operations on each machine according to certain rules (i.e. scheduling rules) or algorithms (i.e. genetic algorithms, ACO algorithms).
In the past, the authors have already introduced Modeling & Simulation (M&S) for supporting production scheduling in real industrial plants (in particular job shop systems). Longo et al. (2003) presents a simulation model for shop orders scheduling in special glass manufacturing; Longo et al. (2006-a) use M&S combined with scheduling rules for production planning within a manufacturing plant devoted to produce high pressure hydraulic hoses. Longo et al. (2006-b) face the scheduling problems (and plant layout optimization) in a wood frames manufacturing process. Finally Longo et al. (2008) combine M&S and genetic algorithms for shop orders scheduling in a real manufacturing plant.

AN APPLICATION EXAMPLE

This section presents the initial results of a research work (developed by authors) that aims at using Modeling & Simulation in a job shop scheduling problem supported by ACO meta-heuristic.
The job shop system manufactures high pressure hydraulic hoses (Longo et al., 2006-a). Each product is made up by a high pressure hose, two adapters and two hydraulic fittings. The production process is made up by 8 different operations:

- Materials preparation
- Fittings stamp
- Hoses cutting
- Hose skinning
- Assembly
- Junction
- Hydraulic hoses Testing
- Final controls and packaging

Two types of Shop Orders (S.Os) can enter the system: normal S.Os and priority S.Os. Usually normal S.Os are scheduled on a 2-weeks time window (each new S.O. enters in the last position of the 2-weeks queue). On the contrary, a priority S.O. can enter the 2-weeks queue in any position at any time (it depends on the priority level of the S.O.). In other words the system allows the *passing* between jobs. Each S.O. has a finite number *m* of operations, one on each machine and it is allowed to work twice a job on the same machine. All the S.Os entered into the system must be necessarily completed.
Machines could not be available during the scheduling period because of the failures. Failures have been modeled by using a negative exponential distribution for both the Mean Time to Failure (MTTF, expressing the time between two consecutive machine failures) and the Mean Time to Repair (MTTR, expressing the time required for repairing the machine). Finally process and set-up's times are considered as stochastic variables each one with a specific distribution form based on historical data. According to these hypotheses it follows that the case analysed belong to the *dynamic-stochastic* scheduling problem because new S.Os arrive during the scheduling horizon and most of the numerical quantities are stochastic.
The simulation model of the job shop system was developed by using the discrete event simulation software eM-Plant. The figure 4 shows the main frame of the simulation model.

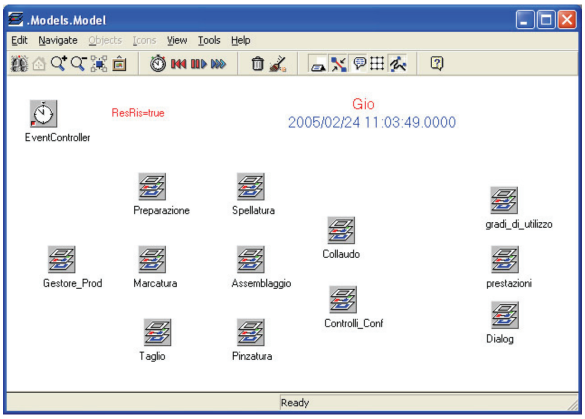

Fig. 4 – Simulation model main frame

Further implementations have been carried out to introduce an Ant System algorithm as support tool for S.Os scheduling. The implementation of the algorithm as well as the interface with the simulation model was created through the programming of specific sub-routines, written using the simulation language Simple++.

Some initial optimizations have been carried out trying to minimize the S.Os flow time and lateness and maximize the fill rate (the last one intended as percentage of S.Os that meet the due date). The flow time reduction was about 10.5% lower than the best result obtained by using the Short Production Time (SPT) scheduling rule. Analogously the lateness reduction was about 13.4% lower than the best solution obtained by using the Due Date scheduling rule. Finally the fill rate improvement was about 5.1% greater than the best solution obtained by using the Due Date scheduling rule.

CONCLUSIONS

The paper presents a state of art about Ants System with particular attention to industrial plants problems. The state of art shows that the main fields of application for ACO algorithms are the routing problems, but in the last years, different applications have been developed to solve industrial plants problems and, in particular, production scheduling problems (job shop, flow shop, open shop). The final part of the paper focuses on the integration of Modeling & Simulation and Ant Theory for supporting real time production scheduling in job shop systems. The authors present some initial results on the development of a simulation model of a real industrial plant (job shop system) supported by an AS algorithm. Further researches are still on going for simulation model accreditation within the real system in order to use it a support tool for real-time production scheduling.

REFERENCES

Blum Christian. Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. Computers & Operations Research 32 (2005) 1565–1591.

Brelaz D. New methods to color vertices of a graph. Communications of the ACM, 22:251–256, 1979.

Bullnheimer B., Hartl R. F., and Strauss C. A new rank-based version of the ant system: a computational study. Technical Report POM-03/97, Institute of Management Science, University of Vienna, 1997. Accepted for publication in the *Central European Journal for Operations Research and Economics*.

Cimino A., Longo F., Mirabelli G., Papoff E. *Shop orders scheduling: dispatching rules and genetic algorithms based approaches.* Proceedings of the European Modeling & Simulation Symposium, 17-19 Settembre, Campora S. Giovanni (CS), Italy.

Colorni A., Dorigo M., Maniezzo V., and Trubian M. Ant system for job-shop scheduling. Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL), 34:39–53, 1994.

Costa D. and Hertz A. Ants can colour graphs. Journal of the Operational Research Society, 48:295–305, 1997.

Curcio D., Longo F., Mirabelli G., Papoff E., *Pharmaceutical Routes Optimization using Artificial Intelligence,* Proceedings of Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 6-8 September 2007, Dortmund, Germany

De Sensi G., Longo F., Mirabelli G., Papoff E., *Ants Colony System for Supply Chain Routes Optimization*, Proceedings of Harbour, Maritime & Multimodal Logistics Modeling and Simulation, October 4th – 6th 2006, Barcelona, Spain.

Dorigo, Maniezzo, Colorni. 1996. "The Ant System: Optimization by a colony of cooperating agents". IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, 1996, pp.1-13.

Dorigo M. 1992. "*Optimization, Learning and Natural Algorithms* (in Italian)". PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT.

Dorigo M. 1993. "Parallel ant system: An experimental study". Unpublished manuscript.

Dorigo M., Maniezzo V., & Colorni A. 1991. *Positive feedback as a search strategy*. Tech. rept. 91-016. Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo M., Di Caro G. 1999. "Ant Colony Optimization: a new meta-heuristic". (a)

Dorigo M., Di Caro G. 1999. "Ant Algorithms for Discrete Optimization". (b)

Dorigo M. and Gambardella L. M. A study of some properties of Ant-Q. In Proceedings of PPSN-IV, Fourth International Conference on Parallel Problem Solving From Nature, pages 656–665. Berlin: Springer-Verlag, 1996.

Dorigo M. and Gambardella L. M.. Ant colonies for the traveling salesman problem. BioSystems, 43:73–81, 1997. (a)

Dorigo M. and Gambardella L. M. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1):53–66, 1997. (b)

Freisleben B. and Merz P. Genetic local search algorithms for solving symmetric and asyimmetric traveling salesman problems. In Proceedings of IEEE International Conference on Evolutionary Computation, IEEE-EC96, pages 616–621. IEEE Press, 1996. (a)

Freisleben B. and Merz P. New genetic local search operators for the traveling, salesman problem. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, Proceedings of PPSN-IV, Fourth International Conference on Parallel Problem Solving from Nature, pages 890–899. Berlin: Springer-Verlag, 1996. (b)

Gajpal Yuvraj, Rajendran Chandrasekharan. An ant-colony optimization algorithm for minimizing the completion-time variance of jobs in flowshops. 2005.

Gambardella L.M., Taillard, E., and Dorigo, M., 1997. "Ant Colonies for QAP". *Tech. Rep. No. IDSIA 97-4*, IDSIA , Lugano, Switzerland.

Gambardella L. M. and Dorigo, M., 1995, "Ant-Q: A Reinforcement Learning approach to the travelling salesman problem". *Proceedings of ML-95, Twelfth International Conference on Machine Learning*, A. Prieditis and S. Russell (eds.) (Morgan Kaufmann, San Francisco, CA) pp. 252–260.

Gambardella L. M. and Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. In Proceedings of the Twelfth International Conference on Machine Learning, ML-95, pages 252–260. Palo Alto, CA: Morgan Kaufmann, 1995.

Gambardella L. M., Taillard E., and Agazzi G. Ant colonies for vehicle routing problems. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, 1999.

Gambardella L. M. and Dorigo M. HAS-SOP: An hybrid ant system for the sequential ordering problem. Technical Report 11-97, IDSIA, Lugano, CH, 1997.

Grassé P. P. 1959. "La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis et cubitermes sp.* La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs." *Insectes Sociaux*, 6:41–81

Gravel M., Wilson L. Price, Caroline Gagn. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. 2001

Leighton F. A graph coloring algorithm for large scheduling problems. Journal of Research of the National Bureau of Standards, 84:489–505, 1979.

Li-Ning Xing ,Ying-Wu Chen, Ke-Wei Yang. Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. 2008

Liao Ching-Jong, Juan Hsiao-Chien. An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. 2005

Longo F., Mirabelli G., Papoff E., *Cape applications in special glass manufacturing*, Proceedings of Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, September 8th – 10th, 2003, pp.385-389 Lviv, Ukraine.

Longo F., Mirabelli G., Papoff E., (2006-a). *Modeling, Analysis & Simulation of Tubes Manufacturing Process and Industrial Operations Controls*, Proceedings of Summer Computer Simulation Conference, July 30th – August 03rd 2006, Calgary Canada.

Longo F., Mirabelli G., Papoff E. (2006-b), *Material Flow Analysis and Plant Lay-Out Optimization of a Manufacturing System*, International Journal of Computing, Vol. 5, Issue 1, 2006, pp. 107-116.

Maniezzo V., Colorni A., and Dorigo M. The ant system applied to the quadratic assignment problem. Technical Report IRIDIA/94-28, Universit´e Libre de Bruxelles, Belgium, 1994.

Merkle D., Middendorf, M., 2000. An ant algorithm with a new pheromone evaluation rule for total tardiness problems.In Proceedings of the EvoWorkshops 2000. In: Lecture Notes in Computer Science,vol. 1803. Springer-Verlag, Berlin,pp. 287–296.

Michel R. and Middendorf M. An island model based ant system with lookahead for the shortest supersequence problem. In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, editors, Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature, pages 692–701. Springer-Verlag, 1998.

Ow PS, Morton TE. Filtered beam search in scheduling. International Journal of Production Research 1988; 26: 297–307

Rajendran C., Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. 2002

Rossi A., Boschi E. A hybrid heuristic to solve the parallel machines job-shop scheduling problem

Rossi A., Dini G. Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robotics and Computer-Integrated Manufacturing 23 (2007) 503–516.

Shyua S.J., Linb B.M.T., P.Y. Yinc. Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. 2004

Stützle T. And Hoos. H. The MAX–MIN ant system and local search for the traveling salesman problem. In T. Baeck, Z. Michalewicz, and X. Yao, editors, Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, pages 309–314. IEEE Press, 1997. (a)

Stützle T. And Hoos. H. Improvements on the ant system: Introducing MAX–MIN ant system. In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, pages 245–249. Springer Verlag, Wien, 1997. (b)

Stützle T. And Hoos. MAX–MIN Ant system and local search for combinatorial optimization problems. In S. Vos, S. Martello, I.H. Osman, and C. Roucairol, editors, Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, pages 137–154. Kluwer, Boston, 1998.

T'kindt V., Monmarché N., Tercinet F., Laüg D.t An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. Laboratoire d'Informatique, Ecole d'Ing_enieurs en Informatique pour l'Industrie, 64 avenue Jean Portalis, 37200 Tours, France, 2001.

Watkins C. J.. Learning with Delayed Rewards. PhD thesis, Psychology Department, University of Cambridge, UK, 1989.

FRANCESCO LONGO took the degree in Mechanical Engineering from University of Calabria (2002) and the PhD in Industrial Engineering (2005). He is currently scientific responsible of the MSC-LES (Modeling & Simulation Center – Laboratory of Enterprise Solutions). He is also Associate Editors of the international journal Simulation: Transaction of SCS. He supports the organization of the most important conferences on M&S (SCSC, EMSS, MAS, etc.). His research interests include modeling & simulation of manufacturing systems and supply chain management, vulnerability and resilience, DOE, ANOVA.

GIOVANNI MIRABELLI was born in Rende (Italy), on January the 24th, 1963. He took the degree in Industrial Technology Engineering from University of Calabria. He is currently researcher at the Mechanical Department (Industrial Engineering Section) of University of Calabria. His research interests include Modeling & Simulation for workstations effective design, work measurement and human reliability.

SABRINA RONDINELLI was born in Tropea (Italy) in May the 9th, 1984. She took her degree in Management Engineering, summa cum Laude, in September 2008 from the University of Calabria. She is currently PhD student at the Mechanical Department of University of Calabria. Her research activities concern the development of simulation models, based on general purpose programming languages and artificial intelligence techniques, for production and Supply Chain planning and management.