# COMPLEX PROJECT SCHEDULING USING MULTI-AGENT METHODS: A CASE STUDY FOR RESEARCH PROJECTS

**Constanta-Nicoleta BODEA**
*Academy of Economic Studies, Bucharest, Romania*
*Piata Romana no. 6, sector 1, Bucharest*
*e-mail: bodea@ase.ro*

**Ileana Ruxandra BADEA**
*Academy of Economic Studies, Bucharest, Romania*
*Piata Romana no. 6, sector 1, Bucharest*
*e-mail: badea_ruxandra@yahoo.com*

**Augustin PURNUS**
*Technical University of Civil Engineering, Bucharest, Romania*
*Lacul Tei Blvd. no. 122-124*
*e-mail: augustin.purnus@spiderproject.ro*

*Management & Marketing*
*Challenges for Knowledge Society*
*(2010) Vol. 5, No. 3, pp. 21-40*

***Abstract.*** *The economy dynamics impacts on the projects constraints, increasing their complexity. In these conditions, the projects could be considered as Complex Adaptive Systems (CAS). The paper addresses the project scheduling optimization problem, when project are seen as CAS. Two different approaches for the project scheduling optimization could be considered: TCPSP (Time-Constrained Project Scheduling), and RCPSP (Resource-Constrained Project Scheduling). The paper focus is on the TCSP with a multi-agent method. The multi-agent approach is chosen because it usually provides better optimization results than deterministic methods. Using as a case study the complex research projects, the paper includes a comparison between two multi-agent methods: Genetic Algorithm (GA) and Ant Colony Optimization (ACO), as a Swarm Intelligence Algorithm. The case study is implemented in MATLAB and further open issues are also presented.*

**Keywords:** project scheduling, multi-agent method, genetic algorithm (GA), swarm intelligence (SI), research projects.

## 1. Introduction

According to a research conducted by Standish Group in 2009, only 32% of projects succeeded, meanwhile 44% did not finish within their initial time limit and budget, 24% completely failed (Standish Group, 2009). Therefore, there is a huge demand for better project planning and scheduling practices. An optimum schedule is not a guarantee for having best results in practice, but it is a requirement for a project to be successful.

In the traditional project management approaches, project planning and scheduling is an exhaustive process, but the situation is changed in the actual accelerated cycles. In modern project management, the planning and scheduling should concentrates on drawing boundaries in order to create a prioritized set of deliverables to be released in iterative phases. The dynamics of the organization structure and management of project-oriented companies (distributed project teams throughout a virtual organization by a network) transform the project planning and scheduling into a vital process.

Different project planning and scheduling methods were developed (Bodea, Elmas, Tanasescu, Dascalu, 2010). The Operational Research (OR) approach provides two major planning techniques: CPM and PERT. Artificial Intelligence initially promoted the automatic planner concept. In order to plan a project, the automatic application of predefined operators is required. However, most domains are not so easily formalized in the form of predefined planning operators. The new AI approaches promote model-based planning and scheduling. In (Bodea, Niculescu, 2006) an agent-based system, named *ResourceLeveler* is presented. The system, an agent-based model for leveling project resources finds a resources scheduling solution which takes into consideration all constraints stemming from the relationships between projects, activity calendars, resource calendars, resource allotment to the activities and resource availability. *ResourceLeveler* was developed in C# as a plug-in for Microsoft Project.

Starting from this viewpoint, we provide solutions to solve CAS scheduling optimization by using adaptive methods: multi-agent methods. Agent technology including agent-based systems and agent-based methods, offers a new way of thinking about many of the classic problems in operations research, including scheduling problems.

## 2. The complex projects as CAS

The nowadays projects complexity recommends them as CAS - Complex Adaptive Systems (Holland, 1995). The study of CAS, a subset of nonlinear dynamical systems, has recently become a major focus of interdisciplinary research in the social and natural sciences. To illustrate the concept of a complex adaptive system, Holland identifies the following major characteristics of CAS: the evolution by aggregation, nonlinearity, anticipation and diversity. The project has the "evolution by aggregation' characteristic, it is not constructed monolithically, having smaller components ("agents"), which are themselves aggregates of still smaller units. These agents are activities, resources influencing each other and impacting differently the overall performance. The behaviour of project agents is not linear, and their interactions are thus not simply additive ones. The projects are characterized by flows to acquire various requirements through networks of agents. Anticipation exhibits two important effects: multiplication (one change produces a change of others) and recycling (feedback loops). And, finally, the project has diversity characteristics; because the project's agents differ from one another (e.g. activities have different times to start, finish, durations and resources assigned)

We can also identify the following CAS three mechanisms:

- **Tagging-entity identification**: Project agents are able to recognize and differentiate among one another, having common characteristics and uniqueness identification elements.
- **Internal models-adaptation**: The internal structure of an agent enables it to anticipate changes in its environment. Project activity agents have time characteristics according to estimated durations and include time buffers that adapt the project schedule to delays caused by project environment.
- **Building blocks**: An agent's internal model is made up of small, reusable modules, thus enabling it to capture a rich set of alternatives with a limited representational vocabulary, e.g. project schedule consists of a list of activities able to switch each other in order to form a more efficient schedule, according to some optimization criteria and specified constraints.

## 3. Scheduling optimization problem for CAS

In order to solve CAS scheduling optimization, the adaptive methods, in a multi-agent implementation are. Agent technology offers a new way of thinking about many of the classic problems in operations research, including scheduling problems. Advanced project management techniques should meet the requirements of distribution and high complexity. Multi-agent methods represents a good approach to optimize project scheduling in a distributed environment The distributed and high complexity of the project environment requires the existence of Activity Agents and Resource Agents representing activities and, respectively, resources in a project. As project team members can be located at different places, resources and operations of a project are distributed by nature. In contrast, classical planning and control of projects is often centralized and all information relevant to the project as a whole should be passed to the project manager (Yan, Kuphal, Bode, 2000).

We find significant differences between multi-agent methods and multi-agent systems/tools. A multi-agent project management system consists of two parts: a small core program that is mainly able to run remote service agents, and a dynamic set of agents residing anywhere in the network. Multi-agent systems (MAS) are a branch in Distributed Artificial Intelligence (DAI). The term *agent* represents a hardware or (more usually) software-based computer system that has the properties of autonomy, social ability, reactivity, and pro-activeness. A stronger notion of agent adopts cognitive notions, such as knowledge, belief, intention, and obligation (Wooldridge & Jennings, 1995). Such MAS examples are GE Job Shop (Baker, 1991) and CASCADE (Parunak, 1988).

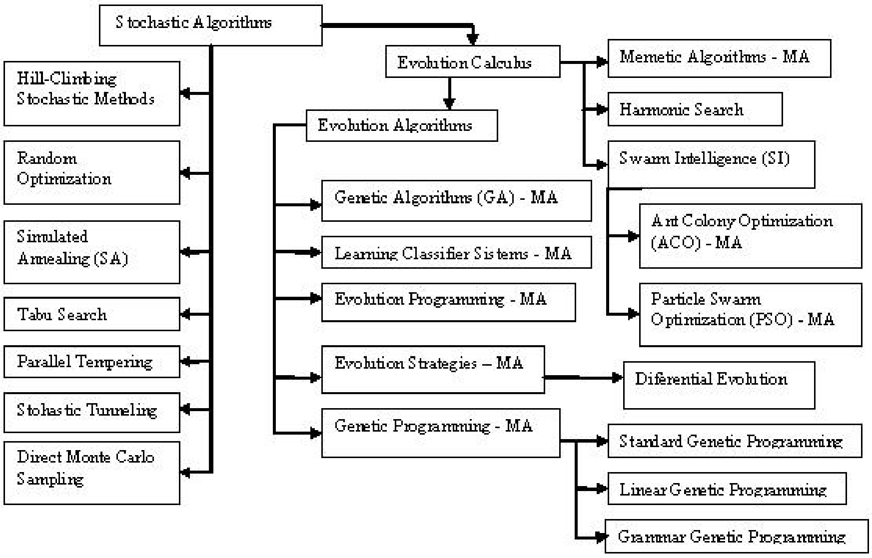
We can define three types of agents for projects: the activity agent, the resource agent, and the service agent (Bodea, Niculescu, 2006). The resource agents and activity agents reside at the site of the project team members who own the resource or implement the activities. The functions of project management are taken by service agents. In general, multi-agent systems for project management try to solve general and complex problems as: multi-objective optimization or project planning and controlling. On the other side, the multi-agent methods are more easily to implement as they refer to a specific management activity (e.g. scheduling, controlling, forecasting), as long as they do not need a specific architecture to exist, only algorithms. In general multi-agent methods are inspired by nature and implement self adaptive behaviour of different intelligent entities: neurons, genes, bees, ants, swarms, etc. In this article, we focus on specific multi-agents: Genetic Algorithm and Swarm Intelligence Optimization, in order to solve a specific problem in project scheduling.

Scheduling problem is known to be NP-complete and has proven to be a difficult task for human planners and schedulers (Simon, 1972), (Hurink, Kok, Paulus, Schutten, 2009). The traditional positioning of scheduling is between planning and execution and it produces decisions on the specific resources, operations, and their timing to perform the tasks (the schedule). Scheduling methods generally fall into two categories: constructive methods and repair methods. Constructive methods incrementally extends a partial schedule until it is complete, meanwhile repair methods iteratively modify a complete schedule to remove conflicts or to further optimize solution. The project scheduling problem is normally characterized by objective functions, features of resources, and pre-emptive conditions (Lee, Kim, 1996). Minimizing of project duration is often used as an objective function, while other objectives such as minimization of total project cost and leveling of resource usage are also considered. For many projects there is a trade-off between project cost and project duration. When a project lags behind its schedule, extra people may be assigned to the job to speed it up. Even for an on-time project there may be opportunities to ‘crash’ the project by hiring personnel or purchasing additional equipment.

The proposed problem to be solved in this article is a project scheduling problem with strict deadlines on the activities; this is meant a *Time-Constrained Project Scheduling Problem (TCPSP).* The TCPSP is a variant of the well-studied *RCPSP (Resource-Constrained Project Scheduling Problem)* which was studied by (Herroelen et al., 1998), (Kolisch, Hartmann, 1999) and (Kolisch, Padman, 2001). However, there are fundamental differences between the time-constrained and the resource-constrained variant. In the first, the deadlines cannot be exceeded and resource profiles may be changed, whereas in the second, the resource availability cannot be exceeded. Moreover, in the TCPSP, a non-regular objective function is considered. Therefore, most existing solution techniques of the RCPSP are not suitable for the TCPSP. Although the TCPSP occurs often in practice, it has been considered only rarely in the literature. (Deckro, Herbert, 1989) give an ILP formulation for the TCPSP and discuss the concept of project crashing. In project crashing, it is possible to decrease the processing time of an activity by using more resources (Kis, 2005), (Li, Willis, 1993). By the nature of their novelty and environment actions, all projects are associated with appropriate levels of uncertainty, and in TCSP we propose the use of time buffers to simulate delays in activities.

In the TCPSP literature, the concept of overtime is rarely analyzed. The TCPSP has, besides to the RCPSP, some relation to the time driven rough-cut capacity planning problem, according to Gademann and Schutten (2005). The rough-cut capacity planning problem is about finding the appropriate capacity level on a tactical decision level, while the TCPSP is on an operational decision level. (Guldemond, Hurink, Paulus, Schutten, 2008). Research has shown that many widely used scheduling techniques (CPM, PERT) are not efficient enough in scheduling linear construction projects with repetitive activities. The management of large projects requires analytical tools for scheduling activities and allocating resources. The following multi-agent methods are suitable for TCSP proposed and will be analyzed as following: PSO and ACO as Swarm Intelligence Methods and GA as Evolution Method (figure 1). In the figure, MA means "support a multi-agent implementation".

**PSO Method (Particle Swarm Optimization).** The general purpose optimization method known as Particle Swarm Optimization (PSO) is due to (Kennedy, Eberhart, 1995), and works by maintaining a swarm of particles that move around in the search-space influenced by the improvements discovered by the other particles (The direction of a particle is then gradually changed so it will start to move in the direction of the best previous position). The advantage of using an optimization method such as PSO is that it does not rely explicitly on the gradient of the problem to be optimized, so the method can be readily employed for a host of optimization problems. This is especially useful when the gradient is too laborious or even impossible to derive. Numerous studies exist on how to make the PSO method perform better. A traditional and easy way of trying to improve the PSO method is by manually changing its behavioural parameters. Various studies have been reported in the literature, for example one by Shi and Eberhart (Shi, Eberhart, 1998) (Eberhart, Shi, 2000) regarding the use of velocity boundaries for the movement of the particles and the choice of the so-called inertia weight which is believed to influence the degree of exploration versus exploitation of the PSO particles.




**Source:** Adapted from Weise (2009).

**Figure 1. Project Scheduling Stochastic Methods**

**ACO Method (Ant Colony Optimization).** The first ACO meta-heuristic, called ant system (Colorni et al., 1991). (Dorigo, 1992) was inspired by studies of the behaviour of ants (Deneubourg et al., 1983), (Deneubourg, Goss, 1989), (Goss et al., 1990). Ants communicate among themselves through pheromone s, a substance they deposit on the ground in variable amounts as they move about. It has been observed that the more ants use a particular path, the more pheromone is deposited on that path and the more it becomes attractive to other ants seeking food. If an obstacle is suddenly placed on an established path leading to a food source, ants will initially go right or left in a seemingly random manner, but those choosing the side that is, in fact, shorter will reach the food more quickly and will make the return journey more often. The pheromone is on the shorter path will therefore be more strongly reinforced and will eventually become the preferred route for the stream of ants. ACO has been applied to several combinatorial problems such as job scheduling, routing optimization in data communication networks and telephone networks (Jones, Bouffet, 2007). An ACO is built in three phases: firstly, the specific problem is mapped in graph form, secondly a tour of solution is created, and lastly the pheromone is updated. Compared to other optimization methods, such as GA, ACO has been found to produce better solutions in terms of computational efficiency and quality when applied to a number of combinatorial optimization problems, such as the TSP - Traveling Salesman Problem (Dorigo, Gambardella, 1997a). ACO has also been successfully applied to scheduling, including the job-shop, flow-shop and resource-constrained project scheduling problems (Bauer et al., 1999), (Colorni et al., 1994), (Merkleet al., 2002), (Stützle, 1998).

**GA Method (Genetic Algorithms).** GA, inspired by the process of biological evolution, have been introduced by Holland (1975) and later developed by Goldberg, Michalewicz and Koza. In contrast to the local search strategies, it simultaneously considers a set of population of candidate solutions instead of one solution. Having generated an initial population new solutions are produced by mating with a specific probability two existing ones (crossover) and/or by altering probabilistically an existing one (mutation). After producing new solutions, the fittest solutions survive and make up the next generation, while the others are discarded. The fitness value measures the quality of a solution, usually based on the objective function value of the optimization problem to be solved. It is important to note that the GA provides a number of potential solutions to a given problem and the choice of final solution is left to the user according to the GA parameters (number of iterations, degree of convergence). In cases where a particular problem does not have one individual solution, for example a family of Pareto-optimal solutions, as is the case in multi-objective optimization and scheduling problems, then the GA is potentially useful for identifying these alternative solutions simultaneously. In the following chapter we will show the effectiveness of genetic algorithms when applied to real world TCPSP.

## 4. A proposed model for the complex project scheduling

In the proposed model, a single project consists of a set A= {0, 1, n+1} of activities which have to be processed. Fictitious activities 0 and n+1 correspond to the “project start” and “project end”. The activities are interrelated by two types of constraints: firstly precedence constraints, force activity j not to be started before its entire immediate predecessor activities comprised in the set Pj have been finished and secondly overall delays for activities should not exceed proposed limits. Resources Rj for activity j are considered sufficient and the only constraint regarding resources is for activity buffers not to exceed an established limit of extra-costs. The objective of TCSP is to find the precedence completion times for all the activities such that the make span of the project to be minimized. Fj represents the finish time for activity j.

A schedule S is given by a vector of Finish times {F1, F2, Fn}. TCSP general form as ILP (Integer Linear Problem) is:

$\mathrm{Min}\ (F_{n+1})$ n+1= final activity of the project (1)
$F_h <= F_j - D_j$ , j=1,..,n+1, h=1,..,j-1, Dj = duration for activity j (2)
$\mathrm{Sum}_j\ [\mathrm{Cost}\ (B\ j)] <= C$, j=1,..,n+1, C = cost limit for buffers regarding resources (3)
$F_j >= 0,\ D_j >= 0, Bj >= 0$ j=1,..,n+1 (4)

To solve this problem we use a similar approach as (Guldemond, Hurink, Paulus, Schutten, 2008). The model is expressed as an integer linear program (ILP). First, we consider the TCPSP with hiring in regular time only, to get a thorough understanding of the problem. Second, we consider the TCPSP with working in overtime, and hiring in regular time and in overtime. Due to the complexity of the problem, we cannot expect to solve the TCPSP via this ILP-model. However, we present the ILP's since the meta-heuristics from Chapter 2, makes use of the ILP formulation to get a feasible solution and to perform a neighborhood search. Relations (1) and (4) will be mixed in one fitness function to minimize a total cost of the project including time and cost resources for activity buffers.

We study the Time-Constrained Project Scheduling Problem (TCPSP), in which the scheduling of activities is subject to mobile deadlines. To be able to meet these deadlines, it is possible to work in overtime or hire additional capacity in regular time or overtime, therefore an extra cost is added. For the TCPSP, a set of n activities, {A1, ..., An}, each job Aj with a finish date Fj and a duration dj, has to be scheduled without pre-emption on a time horizon [0,T]. As a case study we choose a research project with 20 work packages. The research project management is full of uncertainty and complexity. Research has elements of creativity and innovation and accurate prediction of the research outcome is therefore very difficult. It is the research project manager job to manage both the complexities stemming from the culture(s) of researchers/research work and the uncertainties associated with generating research results (Bodea, Dascalu, 2009). Researchers acting safe are more likely to produce conservative and expected results. In order to obtain innovative results, the researchers should have risk-taking behaviour, increasing the probability of failure. This behaviour should be a characteristic at the research system, even at the individual level, it is expected that the researcher will seek to avoid failure. In the majority of research projects, the purpose of project management is also to avoid such failures. It is an apparent conflict between the need for predictability of project output, "on time" and "on budget" and the unpredictability of research outcome and new research opportunities arising in the course of the project. Usually, the quality of output may improve if deviations from plan are allowed.

The researchers ask a large degree of autonomy in their work and democracy in decision making. They co-operate in a research project, but, in the same time, they are strongly competing each others to obtain credit for the results generated in the project, such as: authorship of conference contributions or articles, patents. This competition may lead to conflict between the joint goals of the co-operation and individual goals of researchers. In addition, the relationship between the research project manager and the project participants is characterized by an asymmetric distribution of knowledge where individual researchers know a lot more about the potential – negative and positive – of their research contributions than the project manager does.

To meet these requirements, we use the strategy to put everything into a fitness function representing a cost function for the project and including both time and cost constraints. A fitness function is a particular type of objective function that prescribes the optimality of a solution in a genetic algorithm so that that particular chromosome may be ranked against all the other chromosomes. Optimal chromosomes, or at least chromosomes which are more optimal, are allowed to breed and mix their datasets by any of several techniques, producing a new generation intended to be better.

For the Ant Colony algorithm, the fitness function represents the cost function used to evaluate ant's performances. A good fitness function correlates closely with the algorithm's goal, and yet may be computed quickly. Fitness is typically computed for all members of the population, at each generation of the procedure, and the fitness values for all members are then ranked (first transform) such that the fittest = Rank 1 and the least fit = Rank P, where P is the population size. This ranking (fitness rank) is then used to guide selection of parents for reproduction, with higher ranking parents being selected preferentially. This can be achieved by randomly selecting from the top N or top x% of the ranked individuals, but more commonly some form of rank scaling is performed (second transform) and selection takes place from this modified dataset

In our approach we choose the fitness function based on utility theory, the higher the benefits or performance of the individual, the higher the preference to choose that individual to participate to the next generation. In order to chose the fitness function, several possibilities were compared (Molga, Smutnicki, 2005 ).

a) **De Jong's Function.** The first De Jong's function is one of the simplest test benchmark. It is a continuous, convex and unimodal function (Figure 2). The main disadvantage to apply it in our approach would be to simplify too much the behaviour of complex projects.

f(X)=Sum i (Xi*Xi)

Test area is usually restricted to hypercube
-5.12 <= Xi <= 5.12, i = 1,.., n.
Global minimum f(X) = 0 is obtained for Xi = 0,
i = 1,.., n.

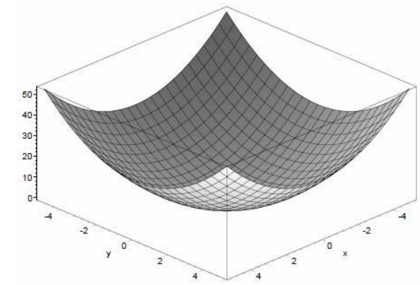


**Figure 2. De Jong's function**

b) **Rastrigin's Function**. Rastrigin's function is based on De Jong's function with the addition of cosine modulation in order to produce frequent local minima (Figure 3). Thus, the test function is highly multimodal. However, the location of the minima is regularly distributed. Multimodal characteristic makes this function to be more useful than DeJong's Function, but still too simple.

f(X) = 10n + Sum i (Xi* Xi-10*cos(2*pi*Xi))

Test area is usually restricted to hypercube
-5.12 · <=Xi ·<=5.12, i = 1,.., n. Its global minimum equal f(X) = 0 is obtained for Xi = 0,
i = 1,.., n.

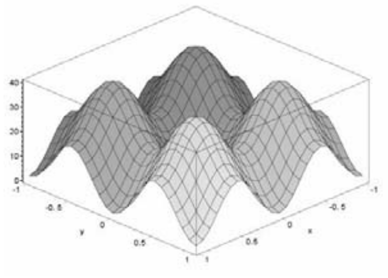


**Figure 3. Rastrigin's function**

c) **Ackley's Function**. Ackley's function is a widely used multimodal test function. The reason not to implement it in this approach is due to the complexity of function, different input data having impact on results (Figure 4).

f(X)= -a* exp(-b*rad (1/n*Sum I (Xi*Xi))- exp (1/n*Sum i (cos (c*Xi)))+a+exp(1)

It is recommended to set a = 20, b = 0.2, c =2 pi.
Test area is usually restricted to hypercube
-32.768 <= Xi <= 32.768, i = 1,.., n. Its global minimum f(X) = 0 is obtained for Xi = 0, i = 1,.., n.

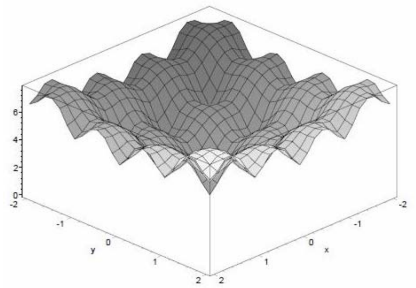


**Figure 4. Ackley's function**

d) **Michalewicz's Function**. This function is used in this approach because it is a multimodal test function (owns *n*! local optima) and has relative complexity. The parameter *m* defines the "steepness" of the valleys or edges. Larger *m* leads to more difficult search. For very large *m* the function behaves like a needle in the haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum). Function has the definition shown in Figure 5.

f(X) = - Sum i {sin(Xi) * [sin (i*Xi$^2$ * 1/ pi) ]$^{2m}$ } (5)

It is usually set m = 10. Test area is usually restricted to hypercube Xi=0,..,pi, i = 1.., n., as seen in the presented figure.

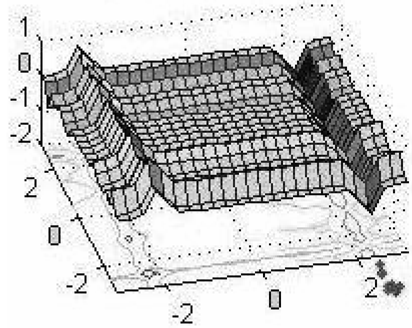



**Figure 5. Michalewicz's function**

**GA Approach.** Each chromosome in the population is in fact a permutation of the buffers associated to project activities (j1, j2, ..., jn). The length of the chromosome will therefore be equal to the number of activities in the schedule. The lower the time buffer – the better the fitness, therefore the fitness is a function depending both on finish time and time buffers for all activities. Regarding the GA operators, we applied crossover in order to generate off-springs from parts of parents' chromosome. If crossover probability is 100%, then all offspring is made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population. Also mutation operator is used to change parts of chromosomes. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed. Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to random search.

**ACO Approach.** The ACO algorithm for Traveling Salesman Problem is modified in order to meet TCSP requirements (relations (1) to (4)), and the fitness function is changed according to Michalewicz's Function general form. The activities are represented in a 2D reference system, with no coding/decoding features. Regarding the ACO parameters, Alpha and Beta are two adjustable parameters that control the relative weight of trail intensity and desirability. If Alpha is big, the ant will seek for other routes. If Beta is big, the ant will follow the shortest route found now, which will cause the premature convergence. Usually, Alpha and Beta are constant, which influence the performance of the algorithm; we can control Alpha and Beta to avoid the premature convergence. Pheromone evaporation is another important parameter used to avoid the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained.

## 5. The case study: complex research project scheduling

In almost all the disciplines of science, the research activities are organized as projects. The initiation of the research project is based on the identification of an original and challenging problem in a specific area of interest where the project proposal team has adequate background and expertise to deal with it. Basic research should be idea driven and should contribute to the existing state of knowledge and lead to further possibilities in terms of basic research as well as applications development. The application/technology development projects should demonstrate specific value addition over the existing state of development in terms of enhanced performance, reduced cost, improved life cycle, environmental impacts, functional aspects, operational considerations and safety reasons. Strategic importance to the nation/society could also be a positive consideration in case of replication, where technology is not freely available.

Although the research as such is an open-ended process, a research project is an endeavor, with precisely defined scope and clear boundaries (project beginning and project end). Accordingly, one should clearly define the specific work to be done as a part of the project and separately list out the outcomes and possible follow up work. Objectives of the research project should be clearly defined. The implementation plan and the research methodology should be aligned to the objectives. The duration of the project is correlated with the work envisaged in the project. The budgeting part should be decided carefully. The number of research staff and their positions/qualifications should be commensurate with the project scope. In order to apply the proposed model to a research project, we consider the activities shown in Table 1 (n = 20).

*Table 1*

**The research project activities**

| **The first 10 activities (with code and name)** | **The last 10 activities (with code and name)** |
|---|---|
| A1. Problem statement | A11. Data analysis |
| A2. Literature survey | A12. First experiments |
| A3. Research objectives | A13. Results analyze, criticize, and explain |
| A4. Research questions | A14. Revisiting the research methodology and data collection |
| A5. Formulating the hypothesis | A15. New experiments |
| A6. Research design | A16. Research report writing |
| A7. Research proposal development | A17. Research communication ate conferences and symposiums |
| A8. Research proposal approval | A18. Research papers in journals |
| A9. Data collection and bibliographical documentation | A19. Research report presentation |
| A10. Data editing and coding | A20. Research report approval |

The following information is associated for each activity:

$F_j$ = earliest finish time for activity j

$B_j$ = time buffer for activity j, considered to be an extra-time due to stochastic factors, and not deterministic

Based on this information can be determined the following:

$S_j$ = start time for activity j = Finish time for activity (j-1)

$D_j$ = duration for activity j, considered to be deterministic = F(j)-F(j-1)

The proposed algorithms, ACO and GA, run on several datasets, each having 20 activities. The single difference between the Datasets consists of different activity buffers randomly generated: ($S_j+D_j+B_j >= F_j$ , $Sj >= F_{(j-1)} - B_{(j-1)}$, j = 1,..,20 ). Table 2 presents the activity-related information.

*Table 2*

**Activity-related information**
**(the activities are ordered by Min Finish Time -Fj)**

| Task Aj | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A145 | A16 | A17 | A18 | A19 | A20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Min Finish Time Fj | 15 | 29 | 43 | 66 | 100 | 131 | 143 | 157 | 174 | 189 | 212 | 227 | 244 | 263 | 284 | 309 | 332 | 361 | 322 | 422 |
| Duration Dj (F(j+1) - Fj) | 14 | 14 | 23 | 34 | 31 | 12 | 15 | 17 | 15 | 23 | 15 | 17 | 19 | 21 | 25 | 23 | 29 | 31 | 30 | 0 |

In both algorithms, Michalewicz Fitness Function is calculated for Xj=(Fj+Bj). For GA approach, the coding/decoding of the solution is done binary by default in code as it is considered less important for this case study and a feature to be improved in a further research. The ACO approach is implemented without any encoding/decoding, the solution being decimal represented.

In order to have comparable results both algorithms run with the same the number of generations/iterations: 100 and the same number of activities = 20. Differences between the two algorithms came from specific parameters. For GA crossover rate is chosen to be maximum: 1, because recombination is needed to be certain and is executed at each iteration. Mutation rate has a low value: 0.003, as it is needed to be rarely executed. For ACO, the alpha parameter is set to 1% and less than beta parameter which is set to value 5%, meaning that ants follow more the shortest route than explore new routes. Also the difference between the two parameters is not significant and gives a low advantage to exploitation than exploration, which is preferred to ACO than GA. The input parameters are given in Table 3.

*Table 3*

**The input parameters**

| GA Parameters | ACO Parameters |
|---|---|
| Encoding/decoding type: binary | Encoding/decoding type: Cartesian (X0Y), decimal |
| Visual representation of solutions: cost vector | Visual representation of solutions: activity graph |
| No. of Generations=100 | iterations = 100 |
| Population Size=20= number of activities | number of nodes=number of activities= 20; |
| Chromosome size binary encoded=16 | number of ants = 16 |
| Crossover Rate= 1.0 | e=.1;%evaporation coefficient |
| Mutation Rate=0.003 | alpha=1;%order of effect of ants' sight<br>beta=5;%order of trace's effect |
| Buffer maximum value =20 time units | Buffer maximum value = 20 time units |

## 2. The simulation results

The simulation environment is MATLAB with the following hardware configuration: Intel Atom 1,60 GHz CPU and 2GHz RAM. The input data consists of two vectors: Min Finish Time (Fj) - presented in the below table and Buffers (Bj) which are randomly generated. Figure 6 presents the ACO simulation code.

The simulation results are shown in Figure 7.

From the presented results, we observe that fitness function has comparable values for minimum criteria (ACO: -3.547, GA: -3.187). Also, the minimum and average fitness graphic does not have significant differences, as shown in the above pictures. Obvious results differences between randomly generated datasets for time buffers were not found.

GA results consists of 20 activity buffers (Bi, i=1,..,20) binary coded on 16 bits (Figure 8a).We choose 16 bits representation because in Table 2 the latest value, F20, has a 16 bit value binary coded. For fitness calculation is used the formula: Xi=Fi+Bi, so Fi and Bi should be consistently represented. After decoding these values, we found one optimum solution for time buffers: 16, 18, 10, 7, 12, 9, 13, 19, 7, 3, 17, 13, 18, 8, 9, 6, 11, 15, 7, 11, having total value: 229.

```
% This ACO is an adaptation of the classical
ACO for Time Constrained
% Scheduling Problem (TCSP) with minimizing
cost for (Buffers+Finish Time)
% Bodea, Badea, Purnus - Bucharest, 2010
% Created and tested under Matlab 6.5 (R13)
[x,y,d,t,h,iter,alpha,beta,e,m,n,el,FinishTime]=a
nts_information;
for i=1:iter
    [app]=ants_primaryplacing (m,n);
```

```
function [cost,f]=ants_cost (m,n,d,at,el,FinishTime);
k=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];% Latest
FinishTime vector
% k will be set to FinishTime+Buffers
for i=1:m
    s=0;
    for j=1:n
        k (j)=at (i,j)+FinishTime (j); % add Buffers to
FinishTime
        s=s-sin (k (j))*(sin (k (j)*k (j)*j/pi))^20; %
```

```
    [at]=ants_cycle (app,m,n,h,t,alpha,beta);
    at=horzcat (at,at (:,1));
    [cost,f]=ants_cost (m,n,d,at,el,FinishTime);
    [t]=ants_traceupdating (m,n,t,at,f,e);
    costoa (i)=mean (cost);
       [mincost (i),number]=min (cost);
    besttour (i,:)=at (number,:);
    iteration (i)=i;
end
subplot (2,1,1);plot (iteration,costoa);
title ('average of cost (distance) versus number
of cycles');
xlabel ('iteration');
ylabel ('distance');
[k,l]=min (mincost);
for i=1:n+1
    X (i)=x (besttour (l,i));
    Y (i)=y (besttour (l,i));
end
subplot (2,1,2);plot (X,Y,'--rs','LineWidth',2,...
              'MarkerEdgeColor','k',...
              'MarkerFaceColor','g',...
              'MarkerSize',10)
xlabel ('X');ylabel ('y');axis ('equal');
for i=1:n
    text (X (i)+.5,Y (i),['\leftarrow node ',num2str
(besttour (l,i))]);
end
title (['optimum course by the length of
',num2str (k)]);
```

```
Michalewicz Function
    end
    f (i)=s;
end
cost=f;
f=f-el*min (f);% elimination of common cost.
```

```
function
[x,y,d,t,h,iter,alpha,beta,e,m,n,el,FinishTime]=ants_i
nformation;
iter=100;% number of cycles.
m=16;% number of ants.
%xoy representation activity lags, randomly
generated
x=rand (20)<.5;
y=rand (20)<.5;
%primary placing for 20 points representing
activities
x=[8 0 -1 2 4 6 3 10 2.5 -5 7 9 11 13 13 -4 -3 -3 1 -
1];
y=[2 4 6 -1 -2 0.5 0 3.7 1.8 1 0 4 3 2 5 2 3 0 3 5];
% take care not to enter iterative points.
n=length (x);% number of nodes=activities
for i=1:n % generating link length matrix.
    for j=1:n
        d (i,j)=sqrt ((x (i)-x (j))^2+(y (i)-y (j))^2);
    end
end
% finish time vector for activities, hardcoded values
FinishTime=[15,29,43,66,100,131,143,157,177,189
,...
    212,27,244,263,284,309,332,361,392,422];
e=.1;% evaporation coefficient.
alpha=1;% order of effect of ants' sight.
beta=5;% order of trace's effect.
for i=1:n % generating sight matrix.
    for j=1:n  if d (i,j)==0 h (i,j)=0; else  h (i,j)=1/d(i,j);
end
    end
end
t=0.0001*ones (n);% primary tracing.
el=.96;% coefficient of common cost elimination.
```

**Figure 6. The ACO simulation code**

ACO results consist of 16 possible optimum solutions, contained by ant's values decimal represented (figure 8b). Each ant has information for 20 time buffers associated to the 20 activities of the project.

Computing the time buffer sums we found very similar values: 210, 198, 218, 192, 210, 210, 210, 210, 204, 210, 210, 210, 200, 210, 210, 211, meaning that ants almost find the same path. Comparing the total time buffer average value for ACO: 195,68 with GA's value: 229, we do not find significant difference as long as the fitness values associated do not differ too much. The maximum total time buffer, assuming maximum buffer values would have been 20 x 20 = 400. The ACO results are a little less than half of this value, and GA is a little above it, but not significantly. This phenomenon can be interpreted in the Gaussian characteristic of randomly generated values within an interval. Table 4 presents the comparative average fitness value for GA and ACO, considering different number of generations.

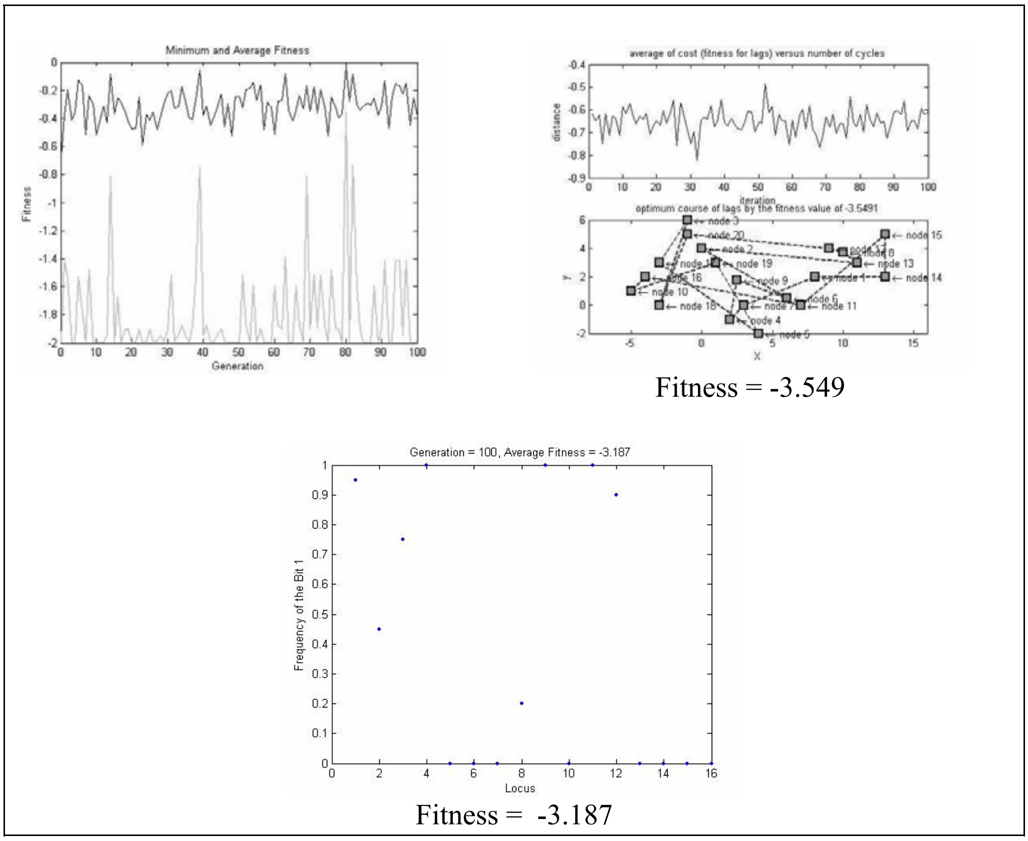



**Figure 7. The simulation results**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

a) GA results: time buffer values (Bj), encoded values for the last generation

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1 | 12 | 7 | 4 | 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 6 | 15 | 14 | 9 |
| 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 4 | 1 | 12 | 5 | 6 | 15 | 14 | 13 | 9 | 7 |
| 14 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 4 | 11 | 7 | 1 | 12 | 9 | 6 | 15 | 5 | 13 |
| 4 | 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 6 | 12 | 5 | 1 | 9 | 14 | 15 | 7 |
| 8 | 13 | 6 | 16 | 18 | 10 | 9 | 17 | 5 | 1 | 12 | 20 | 2 | 3 | 7 | 4 | 11 | 19 | 14 | 15 |
| 17 | 16 | 18 | 10 | 8 | 13 | 9 | 20 | 1 | 12 | 7 | 4 | 11 | 3 | 2 | 5 | 6 | 15 | 14 | 19 |
| 8 | 13 | 20 | 1 | 12 | 5 | 18 | 10 | 9 | 6 | 16 | 19 | 17 | 2 | 3 | 7 | 4 | 11 | 14 | 15 |
| 7 | 4 | 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 6 | 12 | 9 | 14 | 1 | 5 | 15 |
| 16 | 18 | 10 | 8 | 2 | 20 | 19 | 17 | 12 | 5 | 1 | 11 | 3 | 7 | 4 | 14 | 13 | 6 | 15 | 9 |
| 17 | 16 | 18 | 10 | 8 | 4 | 11 | 3 | 2 | 20 | 19 | 7 | 1 | 12 | 9 | 6 | 15 | 14 | 13 | 5 |
| 16 | 18 | 10 | 8 | 4 | 11 | 3 | 2 | 20 | 19 | 17 | 12 | 5 | 1 | 9 | 6 | 15 | 14 | 13 | 7 |
| 1 | 12 | 5 | 18 | 10 | 8 | 16 | 11 | 3 | 2 | 20 | 19 | 17 | 14 | 4 | 7 | 6 | 15 | 13 | 9 |
| 16 | 18 | 10 | 8 | 4 | 11 | 3 | 2 | 20 | 19 | 17 | 12 | 5 | 1 | 9 | 6 | 15 | 14 | 13 | 7 |
| 4 | 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 6 | 12 | 5 | 1 | 9 | 14 | 15 | 7 |
| 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 6 | 12 | 5 | 1 | 11 | 7 | 4 | 14 | 15 | 9 |
| 5 | 1 | 12 | 7 | 4 | 11 | 3 | 2 | 20 | 19 | 17 | 16 | 18 | 10 | 8 | 13 | 9 | 6 | 15 | 14 |

b) ACO results: The ants values

**Figure 8. The results values for time buffers and ants**

*Table 4*

**GA's and ACO's average fitness value for different number of generations**

| Number of Generations | 50 | 100 | 150 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|
| Average Fitness Value by GA | -3.105 | -3.187 | -2.987 | -2.453 | -1.930 | -1.852 |
| Average Fitness Value by ACO | -3.148 | -3.549 | -3.575 | -4.014 | -4.426 | -4.729 |

It is known that on long run, ACO may perform better than the Evolution Algorithms in the early iterations, but it does not appear competitive regarding time when the number of iterations increases. This assumption is demonstrated by successive runs of ACO and GA with increasing number of generations, as it can be seen also in table 4. For GA, the increasing of the number of generations does not give significant better solutions, even worse, and does not decrease significantly time performance.

## Conclusions and further research

In this paper, we studied the CAS behaviour of complex projects, and we observed that an appropriate approach for TCSP problem would be to apply multi-agent methods as Genetic Algorithm and Ant Colony Optimization. From the study case results, it appears that TCSP has acceptable and similar solutions both via GA and ACO approaches. The strength of both algorithms applied in this research is in the parallel nature of their search for optimum solutions.

The differences come only after changing algorithms parameters as number of generations or encoding/decoding type. We observed a little advantage of ACO results quality over GA results especially after increasing number o generations, recommending it to be more robust than GA. Meanwhile after increasing number of generations, ACO runs slower than GA. To avoid this lack of time performance,

ACO can be hybridized with other PSO approaches. Other improvements can be achieved by using different fitness function from Michalewicz's Function to Ackley's Function or Rastrigin's Function. This might have an impact on the performance of the algorithms for specific types of projects. The presented GA can also be modified to have encoding/decoding functions that increases algorithm performance in terms of finding a better solution, and impacts also encoding/decoding time which will be higher.

## Aknowledgements

The results presents in this paper were obtained in the framework of the project ID - 873, financed by Romanian through the Program IDEI. PN CDI II.

## References

Baker, A. (1991), „Manufacturing Control with a Market-driven Contract Net", Ph.D. Dissertation, *Department of Electrical Engineering,* Rensselaer Institute, Troy, NY

Bauer, A., Bullnheimer, B., Hartl, R., Strauss, C. (1999), „An ant colony optimization approach for the single machine total tardiness problem", in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99),* 6-9 July, pp. 1445-1450

Bodea, C.N., Niculescu, C.S. (2006), „Resource Leveling Using Agent Technologies", *The Proceedings of 20th IPMA World Congress on Project Management,* China Machine Press, ISBN 7-111-04703-6, pp. 599-605

Bodea, C.N., Dascalu, M. (2009), *A parameterized web-based Testing Model for Project Management. Proceedings of the Advances in Web-based Learning, ICWL 2009*, Aachen, Germany: Springer, pp. 68-72

Bodea, C.N., Elmas, C., Tanasescu, A., Dascalu, M. (2010), „An ontological-based model for competences in sustainable development projects: a case study for project's commercial activities", *Amfiteatru economic*, no. 27, ISSN 1582-9146, pp. 177-189

Colorni A., Dorigo M., Maniezzo V. (1991), „Positive feedback as a search strategy", Technical Report No 91-016, Politecnico di Milano, Italy, p. 20

Colorni A., Dorigo M., Maniezzo V., Trubian, M. (1994), „Ant system for jobshop scheduling", *JORBEL Belgian Journal of Operations Research, Statistics and Computer Science,* (34), pp. 39-53

Deckro, R.F., Herbert J.E. (1989), „Resource constrained project crashing”, *OMEGA International Journal of Management Science,* 17, pp. 69–79

Deneubourg, J.L., Pasteels, J.M., Verhaeghe, J.C. (1983), „Probabilistic behaviour in ants: A strategy of errors”, *Journal of Theorical Biology* (105), pp. 259–271

Deneubourg, J.L., Goss, S. (1989), „Collective patterns and decision-making” *Ethology and Evolution* (1), pp. 295–311

Dorigo, M. (1992), „Optimization, learning and natural algorithms”, Ph.D. Thesis, *Politecnico di Milano,* Italy

Dorigo, M., Di Caro G. (1999), „Ant Colony Optimization: A New Meta-Heuristic”, *Congress on Evolutionary Computation,* Washington D.C.

Dorigo, M., Gambardella, L.M. (1997), „Ant colonies for the travelling salesman problem”, *BioSystems* (43), pp. 73–81

Eberhart, R.C., Shi Y. (2000), „Comparing inertia weights and constriction factors in particle swarm optimization”, *Proceedings of the 2000 Congress on Evolutionary Computation,* pp. 84-88

Gademann, N., Schutten., M. (2005), „Linear-programming-based heuristics for project capacity planning”. *IIE Transactions,* 37, pp. 153–165

Goss, S., Beckers, R., Deneubourg, J.L., Aron, S., Pasteels, J.M. (1990), „How trail laying and trail following can solve foraging problems for ant colonies”, in Hughes, R.N. (ed.), *Behavioural Mechanisms of Food Selection NATO,* ASI Series, G20, Springer, Berlin

Guldemond T.A., Hurink J.L., Paulus J.J., Schutten J.M.J. (2008), „Time-Constrained Project Scheduling”, Journal of Scheduling, Springer Netherlands, ISSN 1094-6136 (Print) 1099-1425 (Online), vol. 11, no. 2, pp. 134-148

Herroelen, W., De Reyck, B., Demeulemeester, E. (1998), „Resource-constrained project scheduling: a survey of recent developments”, *Computers and Operations Research,* 25, pp. 279–302

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press: Ann Arbor, MI

Holland, J.H., (1995), *Hidden Order: How Adaptation Builds Complexity*, Reading, Addison-Wesley

Hurink, J.L., Kok, A.L., Paulus, J.J., Schutten, J.M.J. (2009), *Time-constrained project scheduling with adjacent resources*, Enschede, The Netherlands

Jones, K., Bouffet, A. (2007), „Comparison of Ant Colony Optimisation and Differential Evolution”, *International Conference on Computer Systems and Technologies* – CompSysTech’07

Kennedy, J., Eberhart, R. (1995), „Particle swarm optimization”, in *Proceedings of IEEE International Conference on Neural Networks,* volume IV, Perth, Australia, pp. 1942-1948

Kis, T. (2005), „A branch-and-cut algorithm for scheduling of projects with variable intensity activities”, Mathematical Programming, 103, pp. 515–539

Kolisch, R., Hartmann, S. (1999), „Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis”, in J. Weglarz (ed.), *Handbook on Recent Advances in Project Scheduling.* Kluwer, pp. 197–212

Kolisch, R., Padman, R. (2001), „An integrated survey of deterministic project scheduling”, *Omega,* 29, pp. 249–272

Lee, J., Kim, Y., (1996), „Search heuristics for resource constrained project scheduling”, *Journal of the Operational Research Society,* 47(5), pp. 678-689

Li, R.K-Y, Willis, R.J. (1993), „Resource constrained scheduling within fixed project durations”, *The Journal of the Operational Research Society,* 44, pp. 71–80

Merkle, D., Middendorf, M., Schmeck, H. (2002), „Ant colony optimization for Resource-Constrained Project Scheduling”, *IEEE Transactions on Evolutionary Computation* 6(4): 333-346

Molga, M., Smutnicki, C. (2005), *Test functions for optimization needs*, Marcin Molga, Czesław Smutnicki, available at http://www.zsd.ict.pwr.wroc.pl/files/ docs/functions.pdf

Parunak, H.V.D. (1988), „Distributed Artificial Intelligence Systems”, *Artificial Intelligence Implications for CIM,* edited by A. Kusiak, Springer-Verlag, Berlin, pp. 225-251

Shi, Y., Eberhart, R.C. (1998), „Parameter selection in particle swarm optimization”, in *Proceedings of Evolutionary Programming* VII (EP98), pp. 591- 600

Simon, H.A., Newell, A. (1972), *Human problem solving*, Prentice-Hall, Englewood Cliffs

Standish Group (2009), *CHAOS Summary 2009*, Report, online available at: http://www1.standishgroup.com/newsroom/chaos_2009.php, accessed on 20-05-2010

Stutzle, T. (1998), „An ant approach for the flow shop problem”, in *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing* (EUFIT '98), 3, pp. 1560-1564

Weise, T. (2009), *Global Optimization Algorithms – Theory and Application*, available at http://www.it-weise.de/, Accesed on 20-05-2010

Wooldridge, M., Jennings, N.R. (1995), „Intelligent agents: theory and practice”, *The Knowledge Engineering Review,* 10 (2), pp. 115-152

Yan, Y., Kuphal, T., Bode, J. (2000), *Application of Multiagent Systems in Project Management*, Int. J. Production Economics, Institute of Production Management and Industrial Information Management Department of Economics and Management University of Leipzig, Germany, pp. 185-197