Multi-objective Scheduling onto Heterogeneous Processors System Using Ant System & Fuzzy Logic Controller

Noureddine LIOUANE[1], Hedi YAHIA[1], Pierre BORNE[2]

[1]Ecole Nationale d'Ingénieurs de Monastir, rue Ibn El Jazzar, 5019 Monastir, Tunisie

[2]LAGIS, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d'Ascq Cedex, France

noureddine.liouane@enim.rnu.tn, pierre.borne@ec-lille.fr

Abstract: In recent years, the static and the dynamic jobs scheduling onto heterogeneous processors present a very well studied problem. Typically the Data Grid Scheduling problem (DGS) has recently become an active research area. The heterogeneous processors scheduling problem (HPSP) can be formulated in several ways and the efficient scheduling of the HPSP on the available resources is one of the key factors for achieving high performance results. Historically, finding an optimal schedule was an NP-hard problem in practical cases; researchers have resorted to devising efficient Heuristics and methods inspired by Nature's Laws. Moreover, the multi-objective scheduling research derives its importance from the need to address the real world of the heterogeneous processors application, which rarely has a single objective function. A schedule that is of a high-quality for one objective function may in fact be quite insignificant for another. Decision makers must carefully evaluate the compromise involved in considering several different criteria in practical scheduling applications. In this paper, we introduce a new hybrid approach that combines ant system optimisation and fuzzy logic concept to facilitate the multi-objective HPSP optimisation, such as the makspean, and the processors workload. Based on the concept of the ant system and fuzzy controller, we automatically control the ant system parameters evolution for the multi-objective HPSP optimisation.

The simulation results indicate that the combination of the ant system approach and the fuzzy controller is not only an efficient metaheuristic tool when we search a multi-objective schedules under constraints but also significantly surpasses other scheduling approaches in terms of quality and solution cost.

Keywords: ant system, fuzzy logic controller, job scheduling, heterogeneous processors, computational grid, large size instances.

Noureddine LIOUANE was born in Kairouan, Tunisia, in 1963. He received the Master degree of science electrical genius in 1988 from the ENSET-Tunis. He received the PhD degree from Ecole Centrale de Lille, France, in 1998. He is currently "Maitre Assistant" in the ENIM-Monastir and Director of the ISSAT-Kairouan. His research is related to the evolutionary optimisation methods for discrete events systems, computer science and operational research.

Hedi YAHIA was born in EL-GFAI-Kairouan, Tunisia, in 1960. He received the Master degree in electrical engineering from ENSET-Tunis, in 1988. Also, he obtained the Doctor degree from "ENIT-Tunis", in1999. He is currently "Maitre Assistant" in the ENIM-Monastir. His main research interests include power electronics system design, scheduling protocol communication and industrial network optimisation.

Pierre Borne : was born in Corbeil, France in 1944, he received the Master degree of Physics in 1967, the Masters of Electronics, of Mechanics and of Applied Mathematics in 1968. The same year he obtained the Diploma of "Ingénieur IDN" (French "Grande Ecole"). He obtained the PhD in Automatic Control of the University of Lille in 1970 and the DSc of the same University in 1976. He became Doctor Honoris Causa of the Moscow Institute of Electronics and Mathematics (Russia) in 1999 and of the University of Waterloo (Canada) in 2006 He is author or co-author of about 200 Journal articles and book chapters, and of 30 plenary lectures and of more than 250 communications in international conferences. He has been the supervisor of 64 PhD thesis and is author of 20 books . He has been President of the IEEE/SMC society in 2000 and 2001 . He is presently Professor "de classe exceptionnelle" at the Ecole Centrale de Lille and director of the French pluriformations national group of research in Automatic Control

1. Introduction

Real world heterogeneous processors scheduling systems are complex and diversified, thereby finding an optimal multi-objective schedule in practical cases is an NP-hard problem and the heuristic approaches must be used.

Conventionally, scheduling problem exists in two forms: Static modes and dynamic modes. In the static case scheduling (Batch-Mode) the jobs are not scheduled as they arise; but they are collected into a set of jobs for the next batch scheduling. While dynamic scheduling case considers a job for scheduling only once. Furthermore, due to the theoretical and practical importance of the scheduling problem, several approaches issue from the directed acyclic graph theory and the soft computing heuristics have been used to examine this problem, [4, 8, 13, 14, 16]. The objective of the scheduling problem is to assign jobs to processors (matching problem), and arrange the execution order of the jobs assigned to each machine (scheduling problem) so that the minimum time of execution is obtained [3]. In the HPSP system considered here, the jobs are assumed independent, i.e. that no constraints of execution exist between the jobs.

For the resolution of the HPSP problems, there exist two approaches: the heuristic methods and the efficient research algorithms including the hybrid heuristics concept. In the static mode of the HPSP, Braun et al. [3] present *eleven* heuristics for matching and scheduling a set of independent jobs onto heterogeneous computing environment, and the goal was to minimise the makespan. Particular modern heuristics approaches have been recently presented for the problem, such as the Tabu Search [1], Simulated Annealing [18], Local Search [15, 11], Genetic Algorithms [5, 7,9,13, 16, 20], Particle Swarm Optimization [2], Fuzzy based scheduling [10]. Moreover, the single objective is not sufficient in practical case and the decision-maker has a priori information and intuition regarding the nature of the optimisation to be performed. For example, the minimisation of the makespan may be of primary importance and the balance of the processor workload and the total workload may be of secondary interest.

Work with other NP-hard problems has shown that the multi-objective solutions found by the heuristics can often be improved by the approaches based on soft computing technique. Essentially, we remark the existence of two grand classes of multi-objective optimisation procedures:

- The Non Pareto approach, that consists in transforming the multi-objective problem to the mono objective problem based on an aggregation operator that mixes the different objectives into a weighted sum. The weighted sum translates multiple objectives into a single objective value for the schedule opportunity [5].
- The Pareto approach is directly based on the Pareto optimisation concept. This approach satisfies two objectives: converge to the Pareto front and also obtain diversified solutions scattered all over the Pareto front [19].Very few researchers have been presented with the multi-objective heterogeneous processors scheduling. The complexity of the problem has a major role to play in this.

In this paper, we propose a new method for finding multi-objective HPSP solutions using metaheuristic based on the Ant system and the Fuzzy Logic Controller (AS-FLC). Moreover the definition of the best bounds and the higher bounds of some criteria are presented. The remainder of this paper is organised as follows. In the next section, we describe the formulation of the heterogeneous processor scheduling problem. A new multi-objective scheduling approach called Ant System and Fuzzy Logic Controller is presented in section 3. The illustrative example and the effectiveness of this approach are illustrated in section 4 and section 5. Finally, section 6 concludes the paper.

2. Problem Formulation

Practically, the heterogeneous processors scheduling problem in computational grids is a multi-objective optimisation problem. More precisely, the HPSP problem may be formulated as follows:

- A set of NJ independent jobs that must be scheduled. Any job has to be processed entirely onto a unique resource, furthermore the pre-emption is not authorised.
- A set of NP heterogeneous processors candidates to participate in the scheduling process.
- An expected processing time $pt_{i,j}$ of each job J_j onto a processor P_i. The $pt_{i,j}$ derived from the $EPTM$ called Expected Processing Time Matrix, where $EPTM_{i,j} = pt_{i,j}$ is expressed in unit of time(*ut*)

Our objective is to minimise the completion time (makespan) and balance the exploitation of the resources effectively. Note that the makespan, total workload and the workload processor are the most important physical parameters of the scheduling problem onto heterogeneous processors.

3. Applying Ant System and Fuzzy Logic Controller to the HPSP

3.1 Construction graph and constraints

Graphically, the HPSP can be represented by the bipartite oriented graph composed of two categories of nodes. A Job is associated to a J node; a processor is associated to a P node. There exists an arc between a J node and a P node if and only if the job J can be executed on the processor P without exceeding its computing capacity.

The cost of the connection J to P is directly linked to the processing time $pt_{i,j}$, given in unit of time (*ut*), of the *j-th* job J_j upon the *i-th* processor P_i. To model the process in a more straightforward manner, we use the construction graph that is derived from the expected processing time matrix $EPTM$. The expected processing time matrix defines the one distance among job J_j and processor P_i.

Table 1. Expected processing time matrix for construction graph

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
P_1	6	4	8	6	3	4	5
P_2	7	5	5	7	5	9	6
P_3	6	8	6	8	7	4	5
P_4	9	7	6	6	5	6	7

Table 2. Example of solution of construction graph

	J_1	J_2	J_3	J_4	J_5	J_6	J_7
P_1		4			3		
P_2			5				
P_3	6					4	5
P_4				6			

With this construction graph, we can transform the HPSP into a travelling ant problem.

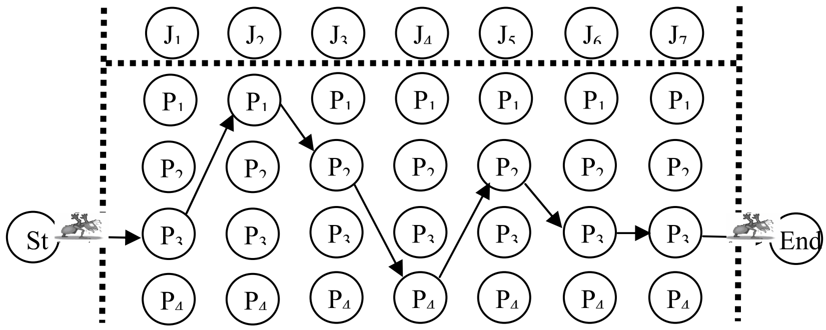

Figure 1. Ant travelling path solution

Above, the construction graph represents a simple solution of the travel of the ant on the path.

Specifically, an ant seeks to travel across the construction graph in such a way that all of the following constraints will be satisfied: one and only one node is visited in each of the columns of the graph. In the rest of this paper, "path" "tour" and "solution" are used interchangeably; a pair of (J_j, P_i) means: Job J_j is assigned to Processor P_i.

3.2 The ant system optimization and fuzzy logic controller

Initially, submitted to application by (Dorigo et al., 1992), the ant system optimisation presents a class of general algorithms of optimisation. The main underlying idea, essentially inspired by the behaviour of real ants, that represents a parallel search of several constructive computational solutions based on the characteristics problem data and on a dynamic memory structure containing information on the quality of previous solutions. Generally, the behaviour of ant system mechanism depends on many unsure parameters, incomplete knowledge of the real ant system attitude and the imprecise information for identification of the relationship between the strategy choice of the parameters and the global behaviour of the ant system heuristics. In our context, we apply the fuzzy logic controller for adapting and tuning the ant system parameters to improve the searching ability in finding the global optimum.

3.3 The ant system scheduling

Typically, ants deposit a chemical pheromone when they move in their environment; they are also able to detect and to follow pheromone trails.

In our case, the pheromone trail describes how the ant system builds the solution of the HPSP problem. On the construction graph, the probability of choosing a branch at a certain time depends on the total amount of pheromone on the branch, which in turn is proportional to the number of ants that used the branch until that time. The probability $P_{i,j}^{k}$ represents the probability of the *k-th* ant to assign a job J_j to an available processor P_i. Each of the ants builds a solution using a combination of the information provided by the pheromone trail $\tau_{i,j}$ and by the heuristic function defined by $\eta_{i,j} = \frac{1}{pt_{i,j}}$.

Formally, the probability of picking that an ant *k-th* will assign a job J_j to the processor P_i is given in equation 1.

$$P_{i,j}^{k} = \begin{cases} \dfrac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{j \in D}(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}} & if\ j \in D \\ 0 & Otherwise \end{cases} \tag{1}$$

In this equation, *D* denotes the set of jobs on the path of the ant, where α and β are parameters that control the relative importance of the pheromone trail versus heuristic measure. Therefore the transition probability is a trade-off between visibility and pheromone trail intensity at the given time.

Adapting α and β parameters automatically permits firstly to improve the ability of ant system in finding the global multi-objective solution and secondly to fine tuning them with respect to the constraints and the criteria described in section 2.

The main idea is to use the fuzzy logic controller to compute new strategy parameter values of the ant system probability $P_{i,j}^{k}$ taking into account the workload and the total workload of the heterogeneous processors.

3.4 Updating the pheromone trail

To allow the ants to share information about good solutions, the updating of the pheromone trail must be established. After each iteration of the ant system algorithm, equation 2 describes in detail the pheromone update used when each ant has completed its own scheduling solution denoted S^k, that represents the length of the ant path. In order to guide the ant system towards good solutions, a mechanism is required to assess the quality of the best solution. The obvious choice would be to use the best makespan $L^{min} = C_{max}$ of all solutions given by a set of ants with minimisation of the criteria C_2 and C_3 described in section 3.6.1.

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{L^{\min}}{L^{k}} & if\ i,j \in S^{k} \\ 0 & Otherwise \end{cases} \tag{2}$$

Where L^k represent the makespan of the solution S^k given by the *k-th* ant.

After all the ants have completed their tours, the trail levels on each node need to be updated. The evaporation factor ensures that the pheromone is not accumulated infinitely and indicates the quantity of the pheromone that is approved over to the next algorithm iteration. The equation 3 represents the pheromone-level-update:

$$\tau_{i,j} = \rho\tau_{i,j} + (1-\rho)\sum_{k=1}^{NA} \Delta\tau_{i,j}^{k} \quad (3)$$

where - ρ is the pheromone evaporation parameter.

- NA defines the number of ants to use in the colony.

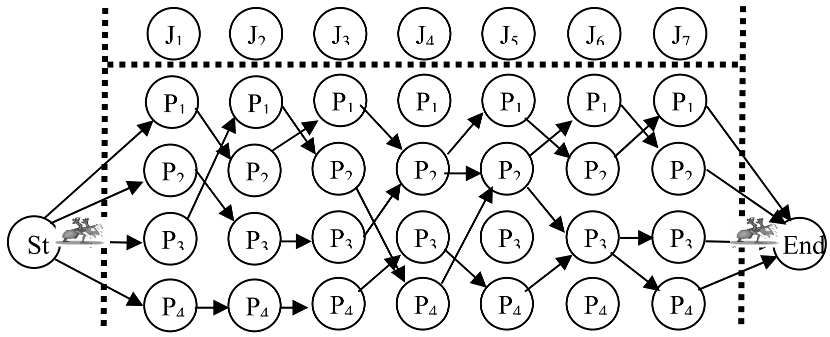

Figure 2. Start construction graph of the HPSP with 4 ants

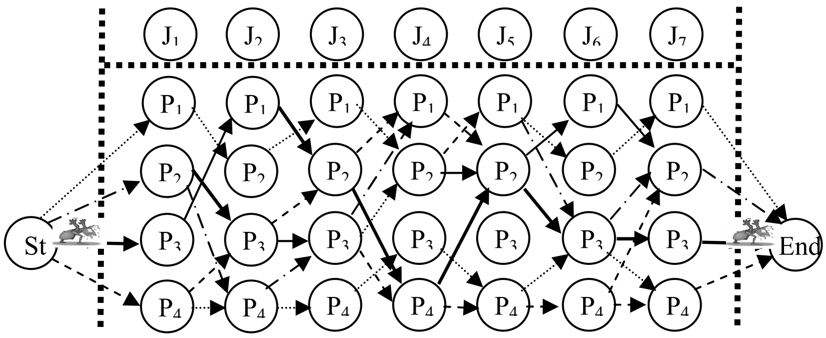

Figure 3. After few iteration the construction graph with 4 ants

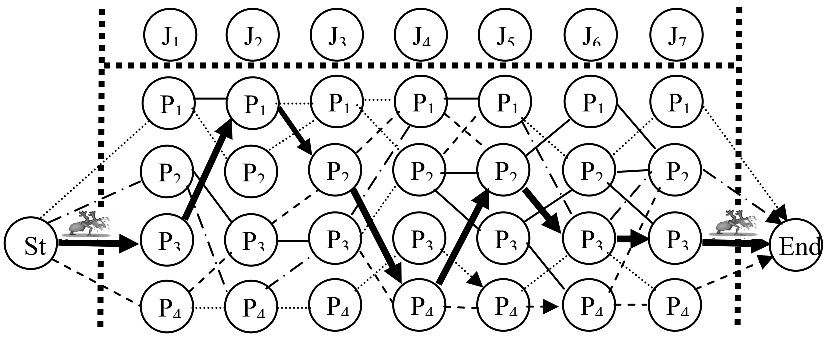

Figure 4. The updating pheromone trail support the path of the best solution and damage the path of the bad solutions. The best path is noted critical path.

3.5 Tabu search application

A simple tabu search was also implemented for this optimisation HPSP problem. The main idea is to allow the ants to build their solutions as described in section 3.1 and then the resulting best solution is taken to a global optimum by the tabu search mechanism. The search process is performed on every critical path, every iteration, so it needs to be fairly fast.

In the case of the HPSP problem, the method is to pick critical path responsible for the C_{max} and verify randomly if any job could be swapped between other processors which would result in a lower makespan. Following their concept, the tabu search considers one problem processor at a time and attempts to randomly swap one job from the problem processor with any other (non-problem) processor. Then the ants are used to generate promising HPSP solutions and the tabu search algorithm is used to avoid entrapment in local minima and to improve these solutions.

3.6 Hybrid ant system and the fuzzy logic controller

Based on the fuzzy system, the fuzzy logic controller appears very useful when the process of the analysis is too complex and where the available information is interpreted qualitatively and approximately. Applying fuzzy system for the adaptation of the α and β parameters regularly does not only enrich the searching ability of the ants mechanism in finding the very good solution but also permits fine tuning them taking into account the criteria (C_2 and C_3) described in section 3.6.1

The fuzzy logic controller is used to automatically adjust the two α and β parameters at different stages in ant system scheduling mechanism.

3.6.1 HPSP Criteria description

Our objective is to find the scheduling that satisfies a multi-objective function under the constraints of the HPSP. In this paper, we consider the balance and compromise between the workload processors and the total workload of the processors.

Let d_j the latest finishing time of the job J_j and w_i characterise the workload of the processor P_i

In this study, the following objectives have been optimised:

- Minimise the makespan or the maximal completion time of the all jobs:

$$C_1 = C_{\max} = \underset{\forall j \le NJ}{Max}(d_j) \cdot ut, \tag{4}$$

- Maximise the workload of each processor, which try to balance the load processors:

$$C_2 = \underset{\forall i \le NP}{Min}(w_i) \cdot ut, \tag{5}$$

- Minimise the total workload of the processors, which represents the total processing time of all processors:

$$C_3 = \sum_{i=1}^{NP} w_i \cdot ut, \tag{6}$$

The criteria C_2 and C_3 give a physical meaning to the HPSP, which refers to the total processing time reduction and to the finding of the workload balance between the set of available processors.

3.6.2 Criteria aggregation by fuzzy logic controller

In this section, two criteria C_2 and C_3 are considered for the multi-objective HPSP problem. C_2 and C_3 are combined by the fuzzy inference. Hence, the fuzzy controller is used to aggregate the criteria. Moreover the decision-maker has to preset the objective criteria and the upper bound values are automatically calculated for every criterion. According to the fuzzy sets theory, the fuzzy controller is based on four principal components:

- The knowledge base: shaping the expertise of the basis rules.
- The fuzzification interface: that transforms the crisp parameters values into fuzzy data.
- The inference process: that represents the criteria aggregation based on the knowledge base.
- The defuzzification interface: that transforms the aggregation results into value parameters action.

3.6.3 Best bounds criteria and fuzzification

A best bound C_i^{BB} for each criterion value is determined via the expected processing time matrix *EPTM* and computed by the following relations.

$$C_3^{BB} = \sum_{\forall j \le NJ} \underset{\forall i \le NP}{Min}(pt_{i,j}) \cdot ut, \tag{7}$$

that represents the minimal total workload of all processors.

$$C_1^{BB} = Max((round(\frac{C_3^{BB} - \varepsilon}{NP}) + 1), \underset{\forall j \le NJ}{Max}(\underset{\forall j \le NP}{Min}(pt_{i,j}))) \cdot ut \qquad , with\ \varepsilon \in [0,1] \tag{8}$$

That represents the best C_{max}

where *round(x)* : round the elements of x to the nearest integer.

$$C_2^{BB} = Max(C_3^{BB} - C_1^{BB}\,(NP-1), \underset{\forall j \le NJ}{Max}(\underset{\forall i \le NP}{Min}(pt_{i,j}))) \cdot ut \tag{9}$$

At the *t-th* iteration, we evaluate the quality of solutions according to the criteria objective. The upper bounds of the C_2^h and C_3^h represent the highest feasible solution obtained by the ant system scheduling mechanism at the *t-th* algorithm iteration. Lastly the fuzzy characterisation of the criteria is given by the membership function μc_i shown in Fig.5.

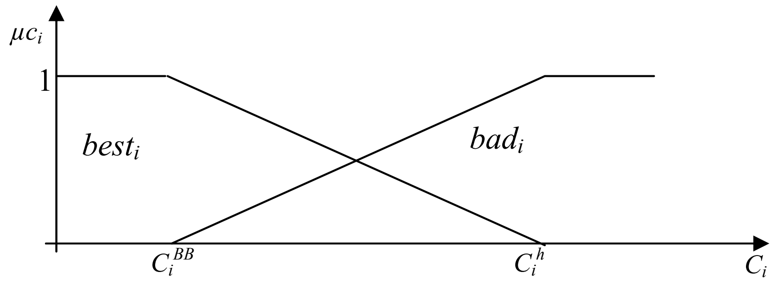

Figure 5. Fuzzy characterisation of the criteria

3.6.4 Knowledge basis and the control ant system parameters

Based on the number of experiments and the expertise in scheduling domain the fuzzy decision rules were constructed for controlling the evolution of the parameters α and β related to the ant system strategy research solutions. The ant system strategy is given by the following basis rules:

If C_2 is $best_2$ and C_3 is $best_3$ then decrease α and increase β for the next iteration ant system algorithm

If C_2 is bad_2 and C_3 is bad_3 then increase α and decrease β for the next iteration ant system algorithm

In order to implement and facilitate the fuzzy logic controller, we note $\mathrm{U} \in \Re^2$ the control ant system vector with $\mathrm{U} = [\alpha \quad \beta]^T$ and $\mathrm{A} \in \Re^{2\times 2}$ the decision strategy matrices that represent the expert decision for increasing or decreasing parameters (α and β) . The equivalent rule bases are given by:

If C_2 is $best_2$ and C_3 is $best_3$ then $\mathrm{U}^+ = A_1 \times U$

If C_2 is bad_2 and C_3 is bad_3 then $\mathrm{U}^+ = A_2 \times U$

In our case the U^+ represents the control vector of the ant system for the next iteration, for example $\mathrm{A}_1 = \begin{bmatrix} 0.75 & 0 \\ 0 & 1.5 \end{bmatrix}$, $\mathrm{A}_2 = \begin{bmatrix} 1.8 & 0 \\ 0 & 0.85 \end{bmatrix}$ can be represent respectively the decision strategy of the first rule and the second rule.

3.6.5 Defuzzification and the inference system

Based on the theory of the fuzzy sets, the fuzzy logic controller concerns the non linear system represented by a set of fuzzy rules of which the consequent part are linear equations. The non linearity is transformed in a weighted sum of these linear state equations.

According to the form of rule bases and the fuzzification phase described previously, the final output of the fuzzy logic controller is inferred as follows:

$$\mathrm{U}^{+} = \frac{\sum_{r=1,2} \mu c^{r} A_{r} U}{\sum_{r=1,2} \mu c^{r}}, \tag{10}$$

where $\mu c^{1} = \min(\mu c_{2.Best}, \mu c_{3.Best})$, and $\mu c^{2} = \min(\mu c_{2.Bad}, \mu c_{3.Bad})$, and r represent the number of fuzzy rules.

3.6.6 The set up parameter values

The set up parameter values used in the ant system scheduling algorithms and the fuzzy logic controller are often very important in getting good results, however the appropriate values are very often entirely problem dependent (Dorigo *et al.* 2002), and cannot always be derived from features of the problem itself . Generally, these parameters are defined by the expertise in the scheduling domain:

- the control vector $U = [\alpha, \beta]^{T}$ summarises how the ant system mechanism exploits the fuzzy controller for adapting the compromise between the pheromone trail and the heuristic information,

- α determines the degree to which pheromone trail is used as the ants build their solutions. The lower the value, the less 'attention' the ants pay to the pheromone trail, but the higher the value, the less exploration the ants perform,

- β determines the extent to which heuristic information is used by the ants,

- the matrices A_1 and A_2 represent how the fuzzy controller increases or decreases the control vector for adapting multi-objective optimisation problems,

- τ_0 is the value to which the pheromone trail values are initialised. Initially the value of the parameter should be moderately high to encourage initial exploration, while the pheromone evaporation procedure will gradually stabilise the pheromone trail,

- ρ is the pheromone evaporation parameter and is always set to be in the range $[0 < \rho < 1]$. It defines how quickly the ants 'forget' past solutions. A higher value makes for a more aggressive search; it tests a value of around 0.5-0.75 to find good solutions,

- *NA* defines the number of ants used in the colony, a low value speeds the algorithm up because less search is done, a high value slows the search down, as more ants run before each pheromone update is performed. Experimentally, a value of 10 appeared to be a good compromise between execution speed and the quality of the solution achieved.

It is interesting to note that for each value of the parameters the ant system scheduling metaheuristic yields a solution. Moreover, its convergence speed depends essentially on the initial number of used ants *NA*.

3.6.7 Building solution steps

The main steps in the strategy of the HPSP by ant system and fuzzy logic controller algorithm are given below:

- Initialise parameters $NA, \alpha, \beta, \tau_0, \rho$.

- Create initial solutions and an empty tabu list of a given size.

In order to generate feasible and diverse solutions, initial ant paths are represented by solutions issued from heuristic rules (MET, MCT, MinMin, MaxMin, FIFO, etc)[3] and random solution paths. Heuristics are used to approximate an optimum as near as possible.

- Repeat the following steps until the termination criteria are met:

- Find *new solution* by the ant system procedure scheduling given in section 3.1.

- Evaluate the quality of the new solution.

- If a *new solution* is improved then the current *best solution* becomes *new solution* *else* If *no new solution* is improved then apply the tabu search optimisation given in section 3.5.

- Add *solution* to the tabu list, if the tabu list is full then delete the oldest entry in the list.

- Apply the updating pheromone trail procedure given in section 3.4

- Compute the upper bound of criteria C_2^h and C_3^h given in section 3.6.3

- Apply the fuzzy logic controller and compute U^+ by the equation 10

- END Repeat

4. Illustrative Example

In order to illustrate the above AS-FLC scheduling methaheuristic, let us consider a HPSP problem composed by 16 jobs J_j *(j=1,2,...,16)* and 4 processors P_i *(i = 1,2,3,4).* Table 3 depict the *EPTM* that gives for each job the expected processing time required by each processor. With $NA = 10; \alpha = 0.25; \ \beta = 0.75; \tau_0 = 0.7; \rho = 0.5$

and $A_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 2 \end{bmatrix}, \ A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}$

Table 3. The expected processing time matrix for 16 Jobs and 4 Processors.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}	J_{16}
P_1	4	3	2	5	2	1	1	2	2	3	3	3	4	6	2	5
P_2	3	2	2	2	1	3	3	3	3	3	2	4	5	6	4	5
P_3	4	5	5	1	4	2	3	1	4	2	4	6	2	6	5	4
P_4	3	2	2	4	2	4	6	5	1	2	6	5	1	6	2	4

In the table 4 we give the values of the best bounds of each criterion; they are computed by the equations 7, 8 and 9.

Table 4. The best bounds criteria for 16 Jobs and 4 processors

C_1^{BB}	*Max*((*round*(33/4)+1) , 6) = 9 *ut*
C_2^{BB}	*Max*((33-9*3) , 6) = 6 *ut*
C_3^{BB}	3+2+2+1+1+1+1+1+1+2+2+3+1+6+2+4 = 33 *ut*

At the first iteration and via the initial solutions generated by the heuristics and the random method, the AS-FLC methaheuristic gives the critical path and the next solution as described below.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}	J_{16}	
P_1	4	3	2	5	2	1	1	2	2	3	3	3	4	6	2	5	
P_2	3	2	2	2	1	3	3	3	3	3	2	4	5	6	4	5	
P_3	4	5	5	1	4	2	3	1	4	2	4	6	2	6	5	4	
P_4	3	2	2	4	2	4	6	5	1	2	6	5	1	6	2	4	
St																	*End*

Figure 6. Initial path issued from the heuristics and random method

Table 5. Initial scheduling issued from the heuristics and random method C_1=10, C_2=7 and C_3 = 34

Scheduling	
P_1	J_{12} >> J_6 >> J_7 >> J_3
P_2	J_{14} >> J_5 >> J_{11}
P_3	J_1 >> J_8 >> J_4 >> J_{10}
P_4	J_{16} >> J_{13} >> J_9 >> J_{15} >> J_2
C_1=10; C_2=7; C_3=34	

The solution given in the table 5 has a C_1=10, C_2=7 and C_3 = 34. The AS-FLC finds that the processor P_4 on the critical path is the problem. To solve this problem and after a few iterations of the travels of the ants (38 in this application example) the AS-FLC metaheuristic suggests to swapping J_1 and J_{16} on the processors P_3

and P_4. The swapping of the job J_1 and J_{16} between the processors P_3 and P_4 gives an amelioration of the scheduling with C_1=9, C_2=7 and C_3 = 33.

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}	J_{16}	
P_1	4	3	2	5	2	1	1	2	2	3	3	3	4	6	2	5	
P_2	3	2	2	2	1	3	3	3	3	3	2	4	5	6	4	5	
P_3	4	5	5	1	4	2	3	1	4	2	4	6	2	6	5	4	
P_4	3	2	2	4	2	4	6	5	1	2	6	5	1	6	2	4	
St																	*End*

Figure 7. Solution path generated by the AS-FLC

Table 6. The best multi- objective scheduling given by the AS-FLC (C_1=9, C_2=7 and C_3 = 33)

Scheduling	
P_1	J_{12} >> J_6 >> J_7 >> J_3
P_2	J_{14} >> J_5 >> J_{11}
P_3	J_{16} >> J_8 >> J_4 >> J_{10}
P_4	J_1 >> J_{13} >> J_9 >> J_{15} >> J_2
C_1=9; C_2=7; C_3=33	

5. Experimental Evaluation

Run by run results Figures 7, 8, 9 depict the results achieved by applying the AS-FLC multi-objective search optimisation, we have evaluated, together, the AS-FLC metaheuristic and initial solution given by the MET, MCT, MinMin and MaxMin heuristics [3] and random method. When generating the *EPTM,* giving the expected processing time of each job J_j on each of the heterogeneous processors P_i we adopted the procedure used in [3] to represent the heterogeneity. In our case, we refer to the inconsistent heterogeneity when no consistency is imposed and the expected processing time matrix is randomly generated without enforcing any rule for consistency. In the context of this work, firstly the processing time $pt_{i,j}$ is chosen Integer randomly generated with uniform distribution over the interval [1..100] and secondly we consider *16* processors and *64, 128, 256, 512* Jobs scheduling . The multi-objective comparisons of the results are based on the makespan C_1, the work load of processors C_2 and the total workload of all processors C_3.

All AS-FLC search optimisation results presented are for 100 iterations runs with 10 the number of ants, and each run was performed 10 times and $\alpha = 0.25;\ \beta = 0.75; \tau_0 = 0.01; \rho = 0.5$

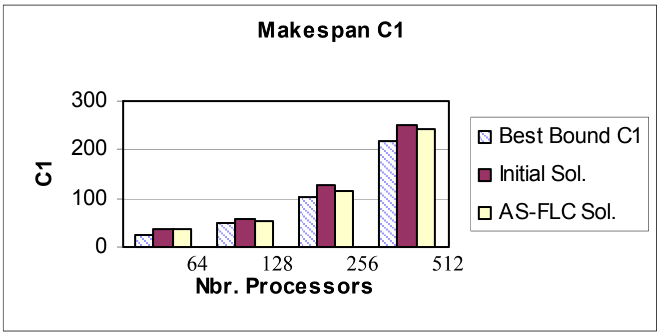

Figure 8. Comparison of the best bound C_1^{BB} and the C_1 given by the initials solutions and AS-FLC approach for inconsistent heterogeneity.

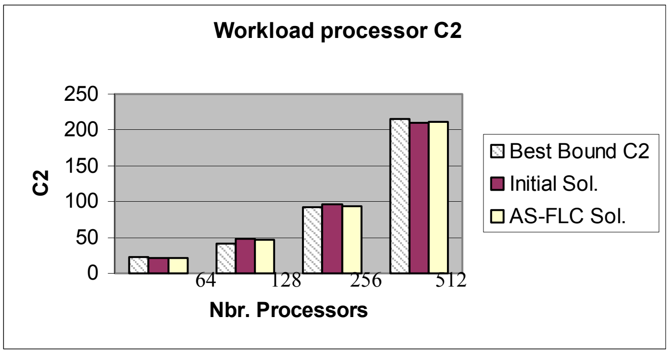

Figure 9. Comparison of the best bound C_2^{BB} and C_2 value given by the initials solutions and AS-FLC approach for inconsistent heterogeneity.

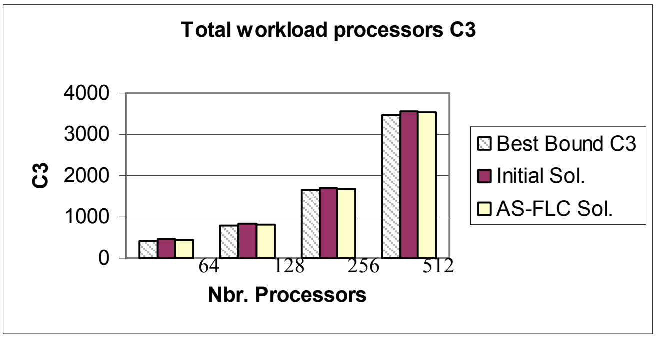

Figure 10. Comparison of the best bound C_3^{BB} and C_3 criterion given by the initials solutions and AS-FLC approach for inconsistent heterogeneity.

In our context it is clear from the uniform results that the AS-FLC approach described is competitive with other hybrid evolutionary algorithms. Via the benefits of the initial solutions given by the heuristics and the computing of the best bounds of the criteria , these results of experiments clearly show the extent to which the ant system rely on the fuzzy logic controller to find very good solutions.

6. Conclusions

In this paper, two related approaches are presented, ant system and fuzzy logic controller for multi-objective optimisation techniques (AS-FLC), and this for solving the heterogeneous processors scheduling problem (HPSP). Via the new manner of the definition of the best bounds of the criteria (makespan, workload of processors and Total workload), the results of the reformulated problems show that the AS-FLC metaheuristic can find an optimised multi-objective solution for many different problems, it can be effectively adapted to deal with the multi-objective HPSP in computational grids.

During the simulation phases, the AS-FLC can find a very good quality of solutions for many different problems; moreover it does persist to find the optimum for some classes of the multi-objective HPSP problems. It is, however, fairly robust as even when an optimal solution is not found, generally very good solutions are provided. In conclusion we believe that we have demonstrated that the ant system, when associated with fuzzy logic techniques, can be successfully applied to the heterogeneous processors scheduling problem in grid application. Hence, illustrative examples are provided to summarise the proposed methodology, and verify its effectiveness.

REFERENCES

1. ABRAHAM, A., R. BUYYA, and B. NATH., **Nature's Heuristics for Scheduling Jobs on Computational Grids**, in The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), India, 2000.

2. ABRAHAM, A., H. LIU, W. ZHANG and T.G. CHANG, **Job Scheduling on Computational Grids Using Fuzzy Particle Swarm Algorithm**, 10th International Conference on Knowledge-

Based and Intelligent Information and Engineering Systems, Lecture Notes on Artificial Intelligence 4252: 500-507, Springer, 2006.

3. BRAUN, T.D. and H. J. SIEGEL, N. BECK, L.L. BÖLÖNI, M. MAHESWARAN, A.I. REUTHER, J.P. ROBERTSON, M.D. THEYS, and B. YAO, **A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems**, Journal of Parallel and Distributed Computing, 61(6): 810-837, 2001.

4. CAO, J., D.P. SPOONER, S.A. JARVIS and G.R. NUDD, **Grid Load Balancing Using Intelligent Agents**, Future Generation Computer Systems, 21(1) : 135-149, 2005.

5. COCHRAN, J.K., S.M. HORNG, and J.W. FOWLER, **A Multi-Population Genetic Algorithm to Solve Multi-Objective Scheduling Problems for Parallel Machines**, Computers and Operations Research, 30: 1087-1102, 2003.

6. DORIGO, M. and T. STÜTZLE, **The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances**, Handbook of metaheuristics, International Series in Operations Research and Management Science, 57: 251–285, 2002.

7. DI MARTINO, V. and M. MILILOTTI, **Sub Optimal Scheduling in a Grid Using Genetic Algorithms**, Parallel Computing, 30: 553-565, 2004.

8. GAO, Y., H. RONG and J.Z. HUANG, **Adaptive Grid Job Scheduling with Genetic Algorithms**, Future Generation Computer Systems, 21(1): 151-161, 2005.

9. HAJRI, S., N. LIOUANE, S. HAMMADI and P. BORNE, **A Controlled Genetic Algorithms by Fuzzy Logic and Belief Functions for Job-Shop Scheduling**, IEEE Transaction on System Man and Cybernetics, 83(1): 812-818, 2000.

10. KUMAR, K.P., A. AGARWAL, and R. KRISHNAN, **Fuzzy Based Resource Management Framework for High throughput Computing,** In Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid, 555-562, 2004.

11. LIOUANE, N., I. SAAD, S. HAMMADI and P. BORNE, **Ant Systems & Local Search Optimization for Flexible Job Shop Scheduling Production,** International Journal of Computers, Communications & Control, 2:174-184, 2007.

12. MAHESWARAN, M., S. ALI, H.J. SIEGEL, D. HENSGEN, and R.F. FREUND, **Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems,** Journal of Parallel and Distributed Computing, 59(2):107-131, 1999.

13. PAGE, J. and J. NAUGHTON, **Framework for Task Scheduling in Heterogeneous Distributed Computing Using Genetic Algorithms**, Artificial Intelligence Review, 4: 415-429, 2005.

14. RADULESCU, A. and A.J.C. VAN GEMUND, **Low-Cost Task Scheduling for Distributed-Memory Machines,** IEEE Transactions on Parallel and Distributed Systems, 13(6): 648-658, 2002.

15. RITCHIE, G. and J. LEVINE, **A Hybrid Ant Algorithm for Scheduling Independent Jobs in Heterogeneous Computing Environments**, In 23rd Workshop of the UK Planning and Scheduling Special Interes Group (PLANSIG 2004), 2004.

16. SHIMIZU, Y., **Multiobjective Optimization for Site Location Problems through Hybrid Genetic Algorithm with Neural Networks**, J. Chem. Engng, Japan, 32: 51-58, 1999.

17. TOPCUOGLU, H., S. HARIRI, and M. WU, **Performance Effective and Low-complexity Task Scheduling for Heterogeneous Computing**, IEEE Transactions on Parallel and Distributed Systems, 13(3):260–274, 2002.

18. YARKHAN, A. and J. DONGARRA, **Experiments with Scheduling Using Simulated Annealing in a Grid Environment,** in 3rd International Workshop on Grid Computing (GRID2002), 232-242, 2002.

19. ZITZLER, E., and L. THIELE, **Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,** IEEE Transactions on Evolutionary Computation, 4: 257-271, 1999.

20. ZOMAYA, A.Y. and Y.H. TEH., **Observations on Using Genetic Algorithms for Dynamic Load-Balancing**, IEEE Transactions On Parallel and Distributed Systems, 12(9):899-911, 2001.