Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 11, Issue 4, February 2013: 60 -67
DOI: 10.53555/tojqi.v4i1.10354

Research Article

# Meta-Heuristic Algorithms For Multi-Objective Subtask Scheduling Problems: Overview

**Uday Kumar K N[1*], Praveena Kumara K.M[2]**

[1,2] Department of Mathematics, Reva Institute of Technology and Management, Bengaluru. Karnataka, India.
E-mail: udayakumarakn@reva.edu.in E- mail: praveenmsk82@gmail.com

## Abstract

Multi objective scheduling problems are considered NP-complete. Hence, calls for a most appropriate heuristic scheduling algorithm to address the issue. Numerous heuristic algorithms have been proposed by researchers in the past, however choosing a most appropriate algorithm for the problem under a particular nature is a difficult task because the algorithms are advanced under numerous assumptions. Therefore, on this observation in this article we discussed four meta- heuristic algorithms to solve multi-objective subtask scheduling problems.
Keywords: Multi-objective scheduling, Subtask scheduling, NP- complete, meta-heuristic algorithms.

## 1. INTRODUCTION

Scheduling in generally defined as the process of assigning number of tasks to the available limited resources with the goal of meeting the recognised objectives. Scheduling in manufacturing industries is defined as the process of allocating 'n' jobs to the available 'm' machines to achieve the time-based objectives such as minimizing the makespan, tardiness, lateness, due date etc. and the cost-based objectives such as production cost, transportation cost etc.[2] As the manufacturing industries play a vital role in contributing to the economy of a nation, the development of an efficient scheduling system to increase the growth rate and productivity becomes the top prior requirement.

The general form of multi-objective scheduling problem is given below:
Max: $F(Y) = [f_1(y), f_2(y), \_\_\_\_\_\_\_\_ f_k(y)]$
Subject to $c_i(x) \leq 0, i = 1,2, \_\_\_\_\_\_ q$
Where,
$Y$ = Decision vector, $Y \in S \in R^n$,
$S$ = Search space and $c_i$ = Constraint
Decision variable $Y^0 \in S$
1. $Y^0$ dominates a decision variable $Y^1 \in S(Y^0 \succ Y^1)$ iff
$$[f_1(y^0), \_\_\_\_\_\_\_\_ f_1(y^1), i = 1,2, \_\_\_\_\_\_\_ k]$$
$$f_1(Y^0) \succ f_1(Y^1), \exists\ i \in \{1,2, \_\_\_\_\_\_ k\}$$
2. $Y^0$ is Pareto optimal iff $\neg\exists Y^1 \in S : Y^1 \succ Y^0$
3. This set $P(S)$ is defined for each Pareto optimal decision variable as

[1]*,[2]Department of Mathematics, Reva Institute of Technology and Management, Bengaluru. Karnataka, India. *E-mail: udayakumarakn@reva.edu.in mail: praveenmsk82@gmail.com

$$P(S) = \{Y^0 \in S / \neg\exists Y^1 \in S, Y^1 \succ Y^0\}$$

4. "Pareto optimal" front $P(F)$ defined for the objective function in $P(S)$ as

$$P(S) = \{F(Y) = f_1(Y), f_2(Y), \_\_\_\_\_\_ f_k(Y) / Y \in P(F)\}$$

5. $Y^0$ is said to be non-dominating if $Y^0$ is non- dominating set by any one of the decision variables in the given set.

A non-dominated solution with respect to the complete search space is called as Pareto Optimal. Since, it's not possible to enhance the decision vectors without disturbing one or more objectives; as a result, multi-objective optimization process produces a large number of non-dominated results. These results are saved in a peripheral archive for computational possessions.

### *1.2.1 CLASSIFICATIONS*

Nagar et al. [1] developed a wide-range of structures depending on the following individualities: nature of problem structure, formation of given shop, process of solution and quality in performance. In his work, he described scheduling difficulties into two types; the first type is the deterministic problem, consists of constraints with limited parameters which can be assured with certainty. The undetermined problems, consists of dispensation situations or structures which cannot be predicted in advance is the second type. Undetermined problems are divided into two types namely fuzzy scheduling and stochastic scheduling. In fuzzy problems, the dispensation situations and limitations are exhibited using uncertain number and in stochastic problems the stochastic capricious are implemented to designate the dispensation restraints and structures. Figure 1.1 represents the classification of the scheduling problems.

Most of the researchers focused on deterministic problems, only few researchers addressed stochastic and fuzzy scheduling problems. Since, in reality most of the scheduling problems consist of uncertainty and multi-objectives, hence in this research we used stochastic and fuzzy methods to solve the scheduling problems.

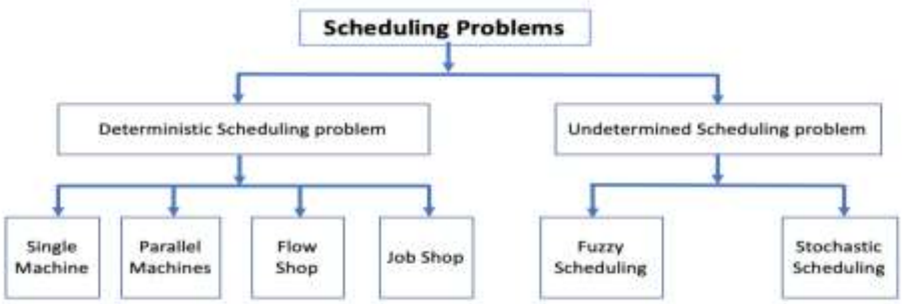


**Figure 1.1 Classifications of scheduling problems**

## 2. Meta Heuristic Algorithms

The increasing complexity in manufacturing industry and growing interest in bio immune and Meta heuristic-based technologies for solving multi-objective subtask scheduling problems has driven the interest for taking this research [5]. It's good to give proper description about the general representation and mathematical model of the proposed heuristic based algorithms. This section explains the basic concepts working mechanism, mathematical representation and the applications of heuristic-based algorithms such as Artificial Immune System (AIS), Particle Swam Optimization (PSO), Subtask Scheduling Algorithm (SSA) and Fuzzy based min-max rule algorithm.

### *2.1 ARTIFICIAL IMMUNE SYSTEM (AIS)*

AIS is an artificial intelligent technique used in scheduling problems from now more than ten years. AIS are inspired by immunology theory of vertebrates, immune functions, working principles and mechanisms of immune system to solve the problems. It acts as a powerful tool to currently existing techniques used for pattern recognition, design, modelling and control. The major application domains of AIS are optimization [3], computer and network security [4], scheduling [6, and 7], data analysis and mining, fault and anomaly detection [8]. The learning process in AIS is the interaction between antibodies and antigens populations which provides a unique self-approach for the inward organizing network structures. Each AIS corresponds to unique number of possible potentials in the presented solutions. The capability to generate novel solutions in a period of short time, inbuilt memory management, robust recognition and self-tolerance are other important features of AIS. These tremendous features made researchers to use AIS to solve multi-objective scheduling problem also. The main goal associated with all scheduling problem is to find a schedule that minimizes the total completion time of the application (makespan). Each solution is an Antibody (Ab); number of libraries will be built each containing a number of genetic strings, these strings being a part of solutions to a scheduling problem. By concatenating threads from individual library an antibody (schedule) is created. Thousand clones of the best specific found were created. The clones were changed and the best clone created is considered as the explanation of the problem.

Immune System (IS) is an in-built defence mechanism acts as a shield to protect all living beings from outside attacks. A biological nature of IS is a robust, adaptive and capable of handling huge mixture of disturbances and uncertainties. This system consists of two major parts namely innate and adaptive immune systems. The innate consists of defences organs like skin and mucus that serves the body from potential threats. These defence organs are supported by phagocytes present in the body tissues and blood. Phagocytes are also sending a signal to identify a threat. All these defence elements are responses to general class of problems and have a restricted and pre-set of responses. If these defence elements of the innate are failed to react to the threats then adaptive is call to react on the infectious agent. The main component of this system is white blood cells (lymphocytes), generated by the bone marrow. Lymphocytes are classified into B-cells, generated in the *bone marrow* and T-cells, generated in the *thymus.* B-lymphocytes produce antibodies and some survives as memory cells. T-lymphocytes function by reacting with other cells. T-cells are divided into helper T-cells, which activates B-cells and destroyer T-cells, which eliminates intracellular pathogens. Stimulated B-cells current bits of the antigens to destroyer T-cells. The immune identification occurs between the region of the receptor and an epitope. Antibodies do not tie to the entire infectious agent, but tie upon many molecules on the surface of infectious agent. **Figure 2.1** shows the immune system mechanism.

Antibodies are bi-functional molecules consists of variable and constant regions. Variations in the variable region provide the IS the power of speed up the adaptation process. Studies on antibody diversity generated during the period of the immune response have shown that the number of somatic mutations in the variable region increases with increase of time. This increases affinity of the antibody for the antigen. High levels of mutations is an important part in IS maturation. Actually, the process of hyper mutation mechanism is completely random; during the process of mutations many will destroy the antigens. The best way the IS addresses this issue by increasing the high-affinity antibodies population. Thus, selection plays an important role in formative high affinity antibodies.

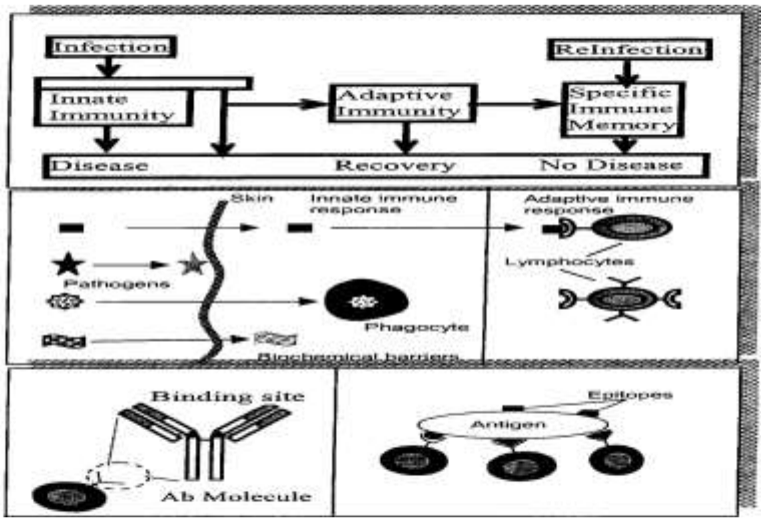


**Figure 2.1 IS function flow (Top), Defense in the immune system (Middle) and Immune recognition (Bottom)**

There exist numerous computational models, based on the mechanism of IS. The most used models are: Bone-marrow models, Negative selection models, Clonal Selection based algorithm model, etc. The selection of these models is completely based on problem exists. The Clonal selection-based algorithm model is given below in detail.

### *3.2.1 CLONAL SELECTION PRINCIPLE*

The Clonal selection principle is based on antigen determined affinity maturation process of B-cells and associated hyper mutation mechanism. De Castro et al. [13] discussed two important principles of affinity maturation in B-cells. The first principle states that proliferation of B-cells is straight relational to the similarity of the antigen that binds it, representative developed the attraction, and more clones are produced. Second opinion states that the mutation under went by the antibody of a B-cell is contrary wise relative to the attraction of the antigen it binds. Using these two principles they developed one of the broadly used Clonal selection based AIS algorithm called CLONALG **(Figure 2.2)**.

In the development of Clonal selection-based algorithm the following aspects were considered; collection and cloning of efficient stimulated cells, affinity mutation and recollection of the clones with high affinity, hyper mutation of cells proportional to their affinity, preservation of the memory cells, decease of non-stimulated cells, production and preservation of diversity. In the algorithm, antibody is an entrant agenda and antigen is a participant agenda which will be the best agenda till that time prompt created by the algorithm. The stepwise working procedure of the algorithm is given below.

Step 1: Generate initial population of antibodies that is generation of potential schedules.

Step 2: Obtain the affinity of the generated antibodies.

Step 3: Generate clones for each antibody; calculation of number clones to be generated depends on the affinity of the respective antibody.

Step 4: Generate maturated cloned population by hyper mutation process.

Step 5: Choose the best clone and discard the remaining clones.

Step 6: Considered the new population as the candidate for the next generation.

This mechanism is continued till the best solution is obtained. This is the best schedule for the optimal problem considered, meeting the pre-defined constraints.

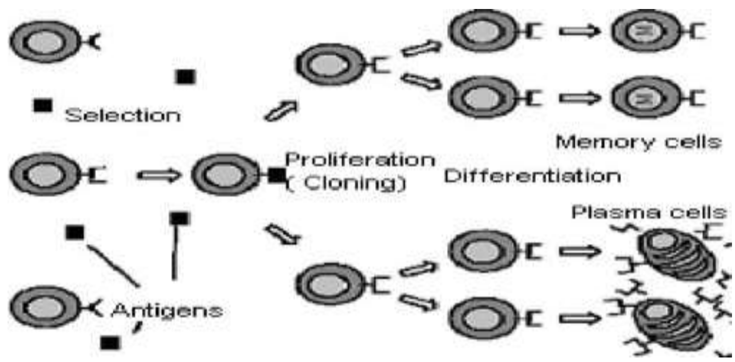


**Figure 3.2 Clonal selection principle**

### *3.3 PARTICLE SWAM OPTIMIZATION (PSO)*

Particle Swarm Optimization (PSO) is an evolutionary population based intelligent technique inspired by the social activities of animals like flock of birds, school of fish etc. PSO technique was proposed by Eberhart and Kennedy [9]. The remarkable features of PSO made the researchers to focus more on this technique are namely optimization through social evolution, its simplicity of usage, it requires simple mathematical operators, and it is low-cost in terms of both memory requirements and time. Here are few applications that have implemented PSO technique are; chemical engineering [11], data mining [12], the voltage control problem, environmental engineering [64], pattern recognition, to solve Scheduling problems [10] and task allocation [15].

The PSO algorithm is analogous to the existing Evolutionary Algorithm. In PSO, the number of particles represents population in a problem space. Particles are randomly initialized. All particles possess fitness values which are calculated by a fitness function, this has to be optimized in every generation. Each particle has its best position denoted as *pbest*, is the best fitness value (result) reached by the particle. The best position among the total group of particles denoted by *gbest*, is the best particle fitness in a total population. The size of the population is problem dependent the most general sizes considered are 20–50 (Hu et al.). In each generation the velocity and particles position will be calculated using Equation 1 and 2, respectively. In Equation1 the first part is the inertia of the prior velocity. The second part is called cognition part, indicates individual thinking, and the last part is social consciousness, indicates cooperation between the particles. The particle new position is obtained by adding up new velocity to the existing position according to Equation 2.

$$V_{i,k+1} = wV_{i,k} + (rand)_1 C_1 (P_i - X_{i,k}) + (rand)_2 C_2 (P_g - X_{i,k}) \tag{1}$$

$$X_{i,k+1} = X_{i,k} + V_{i,k+1} \tag{2}$$

Where,

$$\left[\begin{array}{c} "i = i^{th} particle \\ X_{i,k} = position\ of\ partcle\ 'i\ 'in\ iteration\ 't' \\ V_{i,k} = velocity\ of\ particle\ 'i' in\ iteration 't' \\ p_i = previous\ best\ position\ of\ particle\ 'i' \\ P_g = previous\ best\ position\ among\ all\ the\ particles (g_{best}) \\ w = inertial\ weight - balances\ local\ and\ global\ exploitations\ of\ the\ particles \\ C_1 and C_2 = learning\ factors - controls\ the\ influence\ of\ p_{best} and g_{best} on\ search\ process \\ (rand)^1 and (rand)^2 = random\ numbers - taken\ within\ the\ range\ [0,1]" \end{array}\right.$$

**The summarized process for standard PSO is as follows:**

**Step 1:** Initialize randomly a population of all particles with their random positions and velocities.

**Step 2**: Estimate the suitability ideals of all elements, set *gbest* of each unit like to its existing place, and set *gbest* identical to the place of the best original element.

**Step 3:** Calculate the velocity and position of each particles using Eqs. (1) and (2).

**Step 4:** calculate the fitness values of each particle and compare its existing fitness value with its $p_{best}$ value. If the existing value is better, then update $p_{best}$ with the existing position and fitness value.
**Step 5:** Find out the best particle of the existing population with the best fitness value. If the fitness value is better than $g_{best}$, then update $g_{best}$ with the existing best particle.
**Step 6:** If maximum number of iterations is met, then output $g_{best}$ and its fitness value; if not, go to Step3.

### *3.4 Subtask Scheduling Algorithm*

Due to mass customization, expectations and requirements of customers are diverse. Hence, to offer best services and to satisfy customers, it is more sensible to focus on scheduling of multi-tasks. In general, multi-tasks are divided into two types: Homogeneous and Heterogeneous multi-tasks. The homogeneous task possesses similar characteristics and can be handled by same production processes. In case of heterogeneous task, the characteristics are not similar and have to be processed by range of different processes. The best way to handle heterogeneous multi-task is to decompose a task into number of subtask and processed by aggregated distributed resources[14] (Figure 3.3).
Task decomposition is the process in which a portion of work is divided into two or more task called subtask. The general inspiration to this is the working organization in insect societies. The task decomposition happens in many species like, ants, bees, termites and wasps. Nearly all insects mentioned above concern foraging. Task decomposition is subject to time and costs. Profit occurs through enhancement of individual performance or enhancement through the overall system. The best example to explain task decomposition is collection of nectar in the honey bee. Foragers transport their nectar to bees functioning within the nest, recognized as receivers, who later dump the nectar into cells. Here collection of nectar is therefore decomposed among foragers and receivers. By dividing the task – collection of nectar into two subtasks such as foraging and receiving, task decomposition also causes a novel feature to occur: transportation among forager and receiver. It is the resources that are needed to link each subtask in a decomposed task. Task decomposition requires linking of related tasks. Task is considered as discrete unit of work that has to be completed. The complete task considered in foraging is the collection, repossession, and storage of forage. Evidently, the task is said to be incomplete if forage is collected and brought half the way to its storage. In general, the total completion of work depends on many tasks. Task decomposition is a quite new subject. The decomposition of a task into subtasks and their supply between available resources are of great important, as they significantly affect the production.

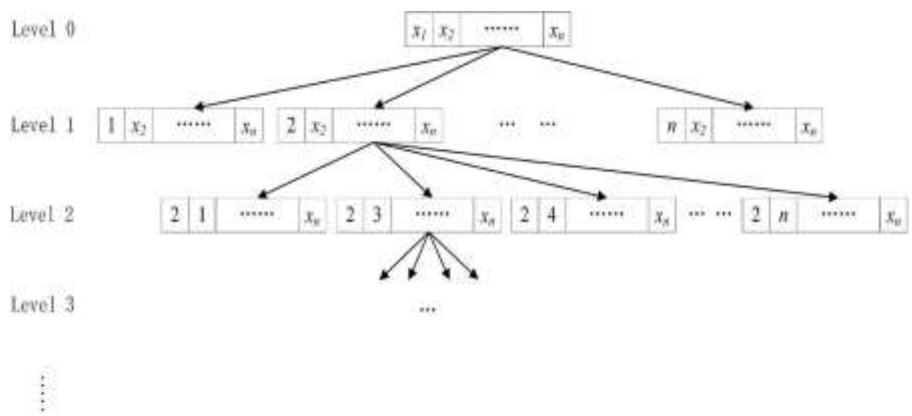


**Figure 2.3 Decomposition of a task into number of subtasks**

A number of articles are published in recent years to advance the effectiveness and usefulness of task decomposition methods by Pi *et al.* [69]. The detailed description about subtask scheduling algorithm is discussed and the basic concept has been given below for developing subtask scheduling algorithm.

#### *3.4.1 BASIC CONCEPT OF SUBTASK SCHEDULING (TASK DECOMPOSITION)*

**1) Partitioning:** The existing most promising task is decomposed into number of subtasks. In each approximation and the left-over solutions is aggregated into the nearby task.
**2) Random Sampling:** A number of samples are chosen randomly from each subtask and the nearby task. Depending on the problem different schemes for sampling can be used. However, the probability of selecting each solution must be positive and when sampling high quality solutions are preferred.
**3) Evaluation of the Promising Index:** The samples objective function values are evaluated for each subtask and nearby task, based on these values the promising indices are determined.
4) **Backtracking:** Based on the promising indices of the subtask and the nearby task, the most promising task in the subsequent approximation is determined. If the most promising index corresponds to a subtask, then that subtask becomes the most promising task in the next approximation. If the most promising index corresponds to the nearby task, then backtracking is performed.

### *3.5 Fuzzy Based Min-Max Rule Algorithm*

Scheduling is a process of assigning limited resources within the constraints to meet the well-defined objectives. A feasible scheduling satisfies the constraints associated with set of tasks and available resources. In a multi-processor scheduling environment, researchers find greater challenges in order to obtain feasible scheduling. Many techniques are proposed to address these challenges like AIS algorithms

Fuzzy logic is a substitute to Boolean logic, in which degree of truth value is used to observe the mode of reasoning, which plays a vital role in decision making ability in uncertainty and imprecision environment. Fuzzy inference consists of three stages as follows:
**1. Input Stage:** Receives input such as target line, completing time, response time etc. and maps these inputs to suitable membership functions.
**2. Processing Stage:** Each suitable rule is invoked and the corresponding results are produced and are joined to give an input to the output stage.
**3. Output Stage:** Converts the joined result back into a specific value.
The membership function depicts curve which enables us to know how each input mapped to a membership value between 0 and 1. The processing stage is based on number of logical rules. The logical rules are stated in the form of IF – THEN rules (in detail if- then rules are discussed in the next session). In order to understand the fuzzy inference technique, it is more relevant to know some basic definitions of fuzzy concept which are given below:

**Min-max algorithm:**
This principle is based on fuzzy set theory that provides a common method for extending crisp domains of mathematical expressions to fuzzy domains. This method generalizes an ordinary mapping of a function to a mapping between fuzzy sets.
Suppose that g is a function from X to Y, and A is a fuzzy set on X defined as

$$A = \{(x_1, \mu_A(x_1)), (x_2, \mu_A(x_2)), \ldots\ldots.(x_n, \mu_A(x_n))\}$$

Then the principle states that the Composition of fuzzy relations R and S

$$SR = SoR\{(x, y), \mu_{SR}(x, z)\}$$

Where, $\mu_{SR}(x, z) = \min\max\{(\mu_R(x, y)), (\mu_S(y, z))\}$

## 4. CONCLUSION

In this article we provided general information and discussed basic concept and working mechanism of heuristic-based algorithms such as AIS, PSO and SSA, also fuzzy concept in scheduling. AIS has attracted many researchers because it has capability to generate new solutions in short period of time, Robust recognition and self-tolerance similarly PSO algorithm has got advantages because of its simplicity of usage and it only requires simple mathematical operators. Subtask scheduling algorithm is useful in case of heterogeneous multiple tasks where characteristics scheduling is not similar and have to be processed by range of different processes. Fuzzy based algorithms are implemented in a multi-processor scheduling environment, where researchers find greater challenges in order to obtain feasible scheduling. Depending on the objective of the problem and nature of the scheduling the algorithms can be selected and implemented.

## REFERENCE

[1] A. Nagar, J. Haddock, and S. Heragu, "Multiple and bicriteria scheduling: a literature survey". Eur J Oper Res, vol.81, pp. 88–104, 1995.

[2] A.M. Costa, P.A. Vargas, F.J. Von Zuben, P.M. França, "Makespan minimization on parallel processors: an immune based approach", in: Proceedings of the Special Sessions on Artificial Immune Systems in the 2002 Congress on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, pp. 115–123, 2002.

[3] D. Dasgupta, and S. Forrest, "Novelty detection in time series data using ideas from immunology", in: Proceedings of the ISCA'96, Reno, Nevada, pp. 19–21, June 1996.

[4] K. Deb, K. Miettinen, and D. Sharma, "A hybrid integrated multi objective optimization procedure for estimating nadir point", In International Conference on Evolutionary Multi-Criterion Optimization, pp. 569–583, 2009.

[5] L.N. De Castro, F.J. Von Zuben, "Artificial immune systems, Part 1, Basic theory and applications", Technical Report, TR-DCA 01/99, 1999.

[6] E. Hart, P. Ross, J. Nelson, "Reducing robust schedules via an artificial immune system", in: Proceedings of the ICEC'98, Anchorage, AK, pp. 464–469, May 1998.

[7] J. Kennedy, and R.C Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, vol. 04, pp. 1942–1948, 1995.

[8] C. u. O. Ourique, E. C. J. Biscaia, and J. C. Pinto. "The use of particle swarm optimization for dynamical analysis in chemical processes". Computers and Chemical Engineering, Vol. 26(12), pp. 1783–1793, 2002.

[9] Pi L., Y. Pan, and L. Shi, "Hybrid nested partitions and mathematical programming approach and its applications," IEEE Trans. Autom. Sci.Eng., pp. 573–586, 2008.

[10] S.H. Russ, A. Lambert, R. King, R. Rajan, D. Reese, "An artificial immune system model for task allocation", in: Proceedings of the Symposium on High Performance Distributed Computing, pp. 3–6, August 1999.

[11] T. Suma and R. Murugesan "Mathematical Model and Heuristic based Subtask Scheduling Algorithm for order scheduling", AIP conference proceedings, April 2019.

[12] T. Sousa, A. Silva, and A. Neves. "Particle swarm-based data mining algorithms for classification tasks". Parallel Computing, vol.30, pp. 767–783, 2004.

[13] B. Yu, X. Yuan, and J. Wang. "Short-term hydro-thermal scheduling using particle swarm optimization method", Energy Conversion and Management, vol. 48(7), pp. 1902–1908, 2007.