

An Experimental Study of Parallel Multi-Objective Genetic Algorithm Models

S. Mishra, S. S. Singh
School of Electronics Engineering
KIIT University, Bhubaneswar, India

B. S. P. Mishra
School of Computer Engineering
KIIT University, Bhubaneswar, India

Abstract- In the real world most of the optimization problems are more than one objective. To address these problems Non-dominated Sorting Genetic Algorithm (NSGA II) is universally referred as a suitable instrument. Nevertheless, problems having many objectives take immense amount of time which leads classical NSGA II to provide solution in tolerable time. In this perspective, parallelization can be an appropriate choice. This paper represents about parallelization of NSGA-II in different models and gives a comparative study by considering a multi-objective rucksack problem. Further, we emphasize on two factors i.e., convergence and time. From the analysis it is clear that cone separation model exhibit better result in comparison to other existing models.

Keywords — PMOGA; NSGA II; Parallel Models; Rucksack Problem

I. INTRODUCTION

In engineering domain majority of the problems are multi-objective in nature, where the objectives are contradictory to each other [1]. In a single objective problem it is easy to find a single solution where as in a Multi-objective Problem (MOP) it is very difficult to converge to a single solution. Hence, the procedure to solve a MOP provides tradeoff solutions, which resides on a Pareto front. This provides more flexibility to the user to select a particular solution. The solutions present on the pareto front are known as non-dominated solution as no other solution are better than them present in the search space by taking into account all objectives at a time. Evolutionary Algorithms (EAs) can find multiple Pareto fronts in a single run. The main objective of Multi-objective Evolutionary Algorithm (MOEA) [1] can be enumerated as below:

- i. The distance between True Pareto font and resultant Pareto font should be minimized.
- ii. Proper distribution of solution in the Pareto font.
- iii. Solution should cover each objective in wide range.

While addressing the Multi-objective problems of higher domain it covers large search space, so the population size increases due to which MOGA consumes huge amount of time to provide the solution. Hence, parallelizing MOGA can be a best solution to address the above problem.

There are two type of parallelism: one is data parallelism and the other one is control parallelism. In case of data parallelism the data is parallelized. In this case a common instruction is operated on a different data set. Whereas in case of control parallelism the instruction is parallelized i.e.

different instructions operate on a common data set in parallel. As in MOEA, a set of solution is achieved, so it opens the possibility of

There exist three basic models for parallelization of MOGA [2, 3, 6]. This paper describes the basic MOGA algorithm i.e. NSGA II parallel implementation and addresses rucksack problem. The result is studied on the basis of two parameter i.e., convergence and time.

Section II describes different models of PMOGA. Rucksack problem and its experimental analysis is discussed in section III and IV respectively. Finally section V describes the conclusion.

II. LITERATURE SURVEY ON EXISTING MODELS

Pragmatic operators of Evolutionary algorithms like crossover, mutation and operations like fitness evaluation can be carried out exclusively on different individuals. Parallization of MOGA has to address the following issues like:

- Selection (globally or locally), Fitness evaluation and mutation;
- Single or multiple subpopulations;
- Crossover in multiple populations.

The Parallelize multi-objective GAs can be figured out into three different categories which are mentioned below.

A. Master-Slave

In this model, one processor known as master processor maintains control over selection, crossover, and mutation. Fitness evaluation is done by other processor known as slave processors. This is done to decrease the overall execution time. Trigger model is a variant of Master slave model as described in Figure 1. Master activates the slaves and each one populates initial solution and performs the fitness evaluation. The calculated fitness value is returned back to the master who performs the other operations. This model uses the perception of in-depth search of solution space through random exploration by the slaves [1].

B. Island Model

In this model each processor is known as a deme. The population is divided among the processors. Then each processor runs the GA independently. The exchange of best individuals between the demes by a process known as migration. The pictorial representation is given in Figure 2. Low communication overhead and higher diversity are the key features of the Island model [3].

C. Cone Separation model

It was suggested by Deb et al. in 2004 [4]. In this model the search space is divided and distributed among the processors. The fitness space is also partitioned in to cones. In this model they also used the concept of frequent normalization and renormalization to handle the individuals. This approach is established with NSGA II [4, 6]. The pictorial representation is given in Figure 3.

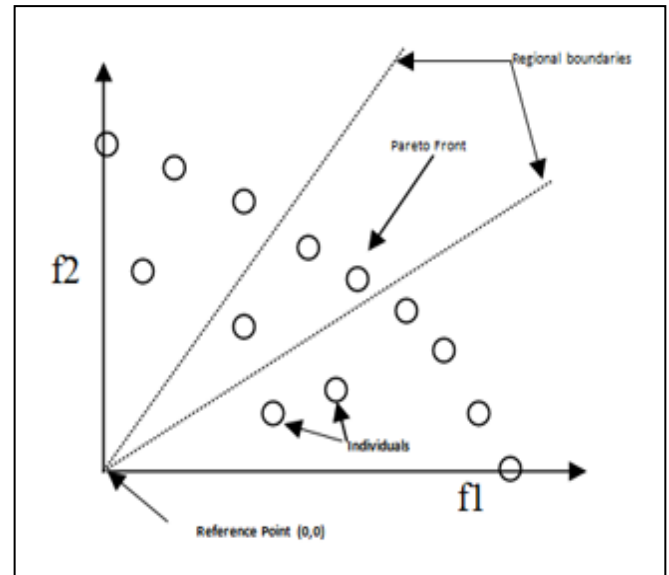


Fig. 3. Cone separation Model

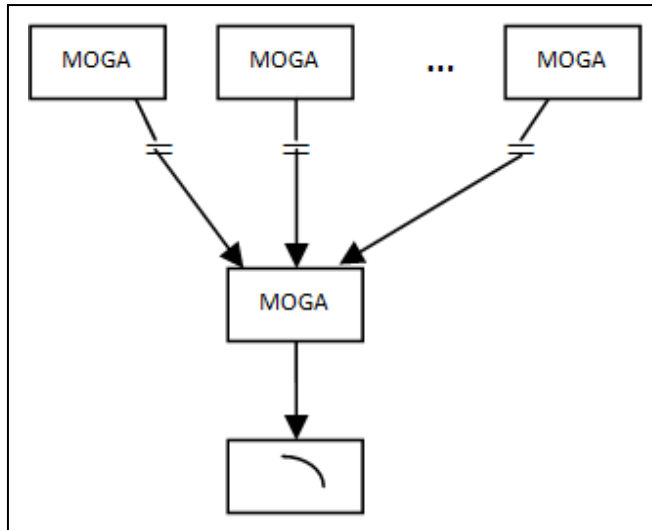


Fig. 1. Master-slave Model

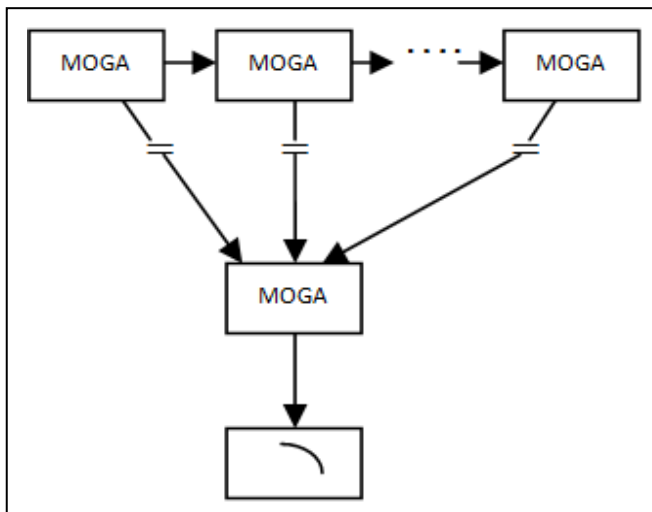


Fig. 2. Island Model

III. MULTI-OBJECTIVE RUCKSACK PROBLEM

The Rucksack problem is a well known combinatorial optimization problem due its NP-hard and realistic nature. Many papers can be founded in the literature about multi-objective Rucksack problem or knapsack problem and about the algorithms proposed for solving them [5, 7]. 0/1 knapsack problem is a variant of Rucksack Problem. Keeping the capacity constraints in the knapsack the profit has to be maximized. By making the number of knapsack variable the problem can be converted to a multi-objective problem. Multi-objective 0/1 knapsack problem can be formulated by (1) and (2)

Let us, consider t = set of items and k = set of knapsacks

$P_{a,b}$ = Profit earned from item b with respect to knapsack a ,
 $w_{a,b}$ = Weight of item b with respect to knapsack a ,

C_a = capacity of knapsack a .

Objective: Uncover a vector $x = (x_1, x_2, \dots, x_t) \in \{0, 1\}^t$ such that

$$\forall i \{1, 2, \dots, k\} \tag{1}$$

and for which $f(x) = (f_1(x), f_2(x), \dots, f_k(x))$ is maximum, where

$$f_i(x) = \sum_{b=1}^t P_{ab} \cdot x_b \tag{2}$$

and $x_b = 1$ iff item b is selected.

The Following suggestions in [7], the knapsack capacities are set to half the total weight according to the corresponding knapsack as indicated in (3).

$$C_i = 0.5 \sum_{b=1}^t w_{ab} \tag{3}$$

IV. EXPERIMENTAL ANALYSIS

This part describes the experimental analysis of the problem.

A. Experiment has done in the following setup

TABLE I. EXPERIMENTAL SETUP

Programming Language	System	Speed of the Processor	RAM	OS
C	I7 , 8 cores	1.4 Ghz	2 GB	Linux

B. Experimental result

To find out the solution to the problem, the experiment is conducted with a very well-known multi-objective 0/1-knapsack problem. Problem is addressed by all the 3 models by using two processor and four processors and it is compared with the result using single processor. The Figure 4 explains the convergence result in trigger model. From that Figure, it can be seen that in a single processor the profit is maximum than two processor and four processors. The convergence quality is inversely proportional to the number of processors.

The same thing we get when dealing with cone separation model (Figure 5.) and island model (Figure 6.). Figure 7. and Figure 8. explains the comparative analysis on different models having two processors and four processors respectively.

While running the experiment it has been observed that in single processor every independent run terminates due to the first termination condition (i.e., average profit greater than 20000). In case of two processor cone separation 10% of the independent run stop due to the first termination condition and rest terminates due to the stop of the Pareto movement (1st termination condition). In all other models, irrespective of the number of processors, algorithm terminates due to the steady state of the Pareto. So it can be concluded that two processor cone separation model is relatively better than any other model, irrespective of the number of processors. Since single processor terminates only because of the average profit exceeds the upper bound (i.e., 20000) we can assume that the uni-processor could have converged more, leading to much more better results. Again the analysis has been done by considering one of the basic parallel parameter, i.e., time. TABLE III explains the time taken for convergence of Pareto front in single processor, two processor and four processors by using the cone separation model, island model and trigger model respectively. In TABLE III, the column one explains about different models taken, column two explains about number of processors taken in each model and column three represents the time taken by the models to converge to the Pareto front.

The Figure-9 explains the speed up obtained in different models by increasing the number of processors.

TABLE II. PARAMETER SETTING

Processor	Population size			Crossover rate	Mutation rate	Termination condition
	1	2	3			
	300	200	100	0.9	0.15/bit	No. of generation or Profit > 20,000

TABLE III. TIME TAKEN FOR CONVERGENCE OF PARETO WITH DIFFERENT FRONT BY DIFFERENT MODELS

Model	No of Processor	Time Taken (in sec)
Single Processor	1	38.159
Cone separation	2	11.242
	4	15.644
Island	2	13.7107
	4	9.0811
Trigger	2	10.4696
	4	9.1595

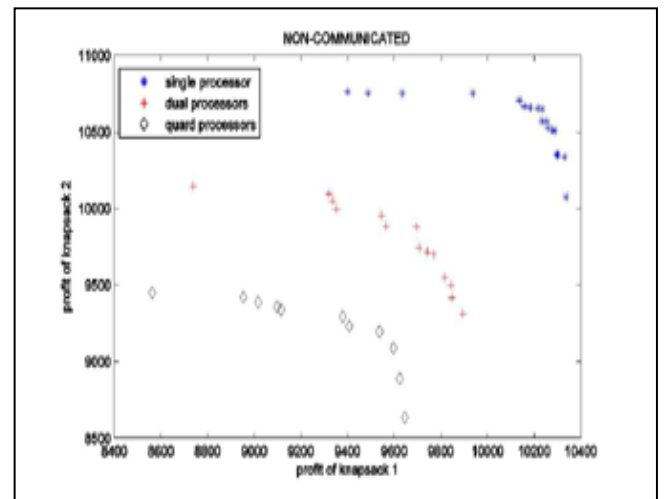


Fig. 4. Convergence result in trigger model using different processors.

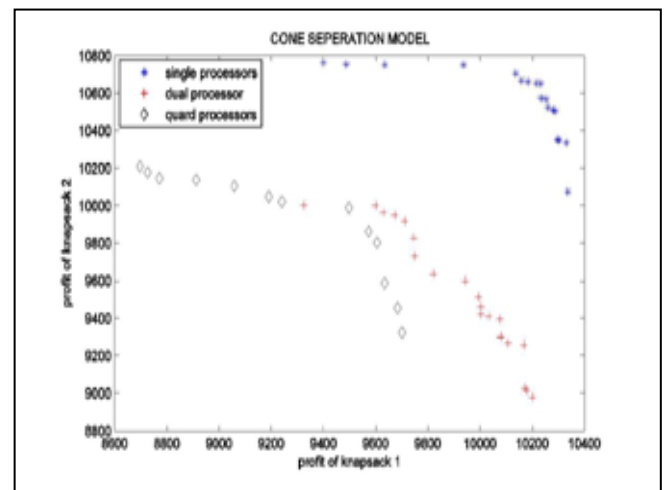


Fig. 5. Convergence result in cone separation model using different processors.

By analyzing it has also been found that the time taken by the single processor is more due to its characteristics of constant movement towards the true Pareto front. All other models take less time than the single processor but the quality of the convergence deteriorates. Hence there is a bit tradeoff between the time and the convergence. It is clear that if we consider the time constraint the island model takes

less time for obtaining the Pareto front than any single and parallel models.

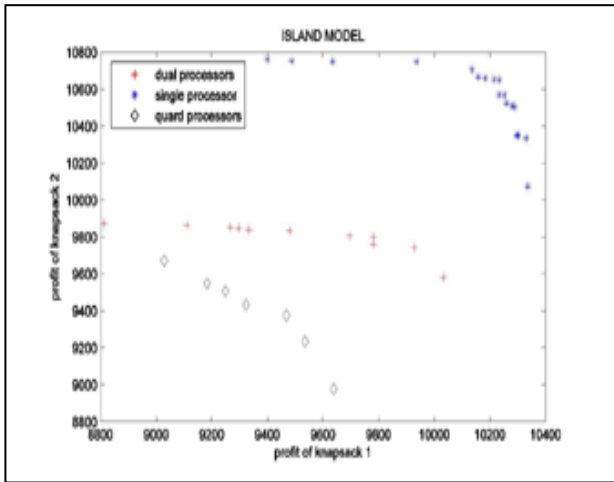


Fig. 6. Convergence result in island model using different processors

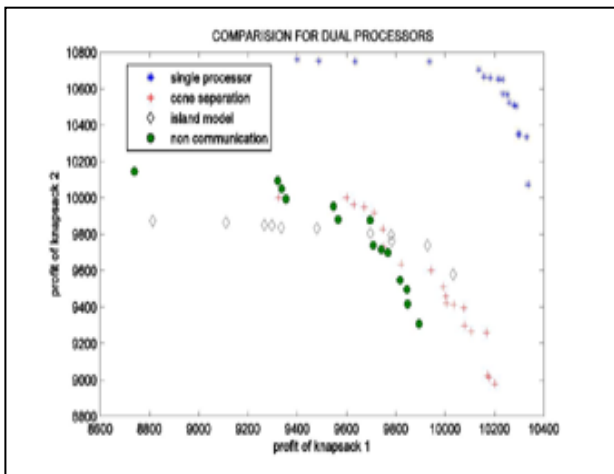


Fig. 7. Convergence result of two processors with different models

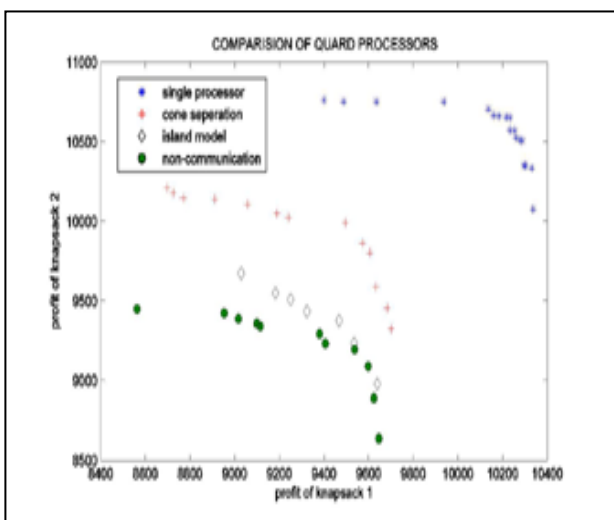


Fig. 8. Convergence result of four processors models

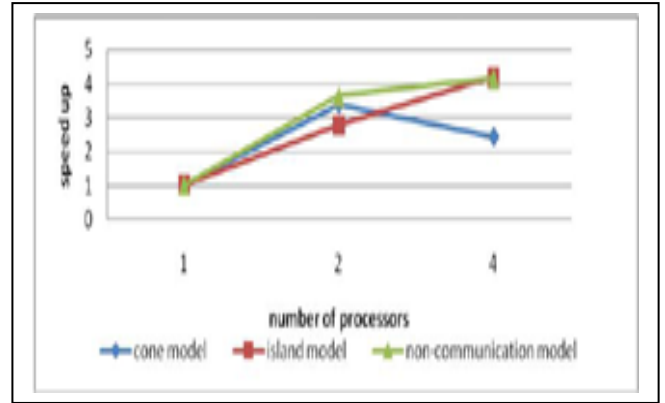


Fig. 9. Speedup analysis among different models

V. CONCLUSION

This paper presents a comparative analysis on different models of PMOGA by implementing it on 0/1 knapsack problem. It is concluded that, cone separation method provides an opportunity to explore the pareto font in parallel. Over all it is found that, cone separation model has the ability of better convergence than any other model. While considering the time parameter it is concluded that, single processor takes more time than any other parallel processing models but having better convergence.

REFERENCES

- [1] Al-Somani, T & Qureshi, Kalim (2000), Reliability Optimization Using Genetics Algorithms", M. Sc. Thesis, Saudi Arabia: King Abdul-Aziz University.
- [2] Branke, J., Schmeck, H., Deb, K. & Reddy, M. S. (2004), "Parallelizing Multi-objective Evolutionary Algorithms: Cone Separation", a congress on evolutionary Computation, Vol. 2, pp. 1952-1957.
- [3] Cantu-Paz, E. (1998), "A Survey of Parallel Genetic Algorithms", Calculateurs Paralleles, Vol.10, No. 2, pp. 141-171.
- [4] Deb, K., Pratap, A., Agrawal, S. & Meyarivan, T. (2002), "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182-197.
- [5] Grosan, C. (2003), "How to compare the multi-objective evolutionary algorithms performances?", Zilele Academice, Clujene.
- [6] Streichert, F. B., Ulmer, H. & Zell, A. (2005), "Parallelization of Multi-objective Evolutionary Algorithms Using Clustering Algorithms", In Evolutionary Multi-Criterion Optimization ed. by Coello Coello, Carlos, Hernández Aguirre, Arturo, Zitzler, Eckart, Springer, Vol. 3410, pp. 92-107.
- [7] Zitzler, E., Deb, K. & Thiele, L. (1999), "Comparison of multi-objective evolutionary algorithms: Empirical results", Institution Swiss Federal Institute of Technology (ETH), Zurich.