Solving Scheduling Problems with Genetic Algorithms using a Priority Encoding Scheme

J.L. Subirats [1], H. Mesa [1], F. Ortega-Zamorano [2], G.E. Juárez [3], J.M. Jerez [1], I. Turias [4] and L. Franco [1]

(1) Departamento de Lenguajes y Ciencias de la Computación, ETSI Informática, Universidad de Málaga (Spain), (2) School of Mathematics and Computer Science, Yachay Tech, San Miguel de Urcuquí, (Ecuador), (3) Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán (Argentina), (4) Departmento de Ingeniería Informática, Universidad de Cádiz (Spain).

ABSTRACT

Scheduling problems are very hard computational tasks with several applications in multitude of domains. I this work, we solve a practical problem motivated by a real industry situation, in which we apply a genetic algorithm for finding an acceptable solution in a very short time interval. The main novelty introduced in this work is the use of a priority based chromosome codification that determines the precedence of a task with respect to other ones, permitting to introduce in a very simple way all problem constraints, including setup costs and workforce availability. Results show the suitability of the approach, obtaining real time solutions for tasks with up to 50 products.

GENETIC ALGORITHMS

A genetic algorithm (GA) generates solutions to optimization problems using operators inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

Candidate solutions to the optimization problem play the role of individuals in a population, while a fitness function determines the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators.

Fitness function

$$fitness = Total\ execution\ time + \sum p(j_i) \times Delay(j_I)$$

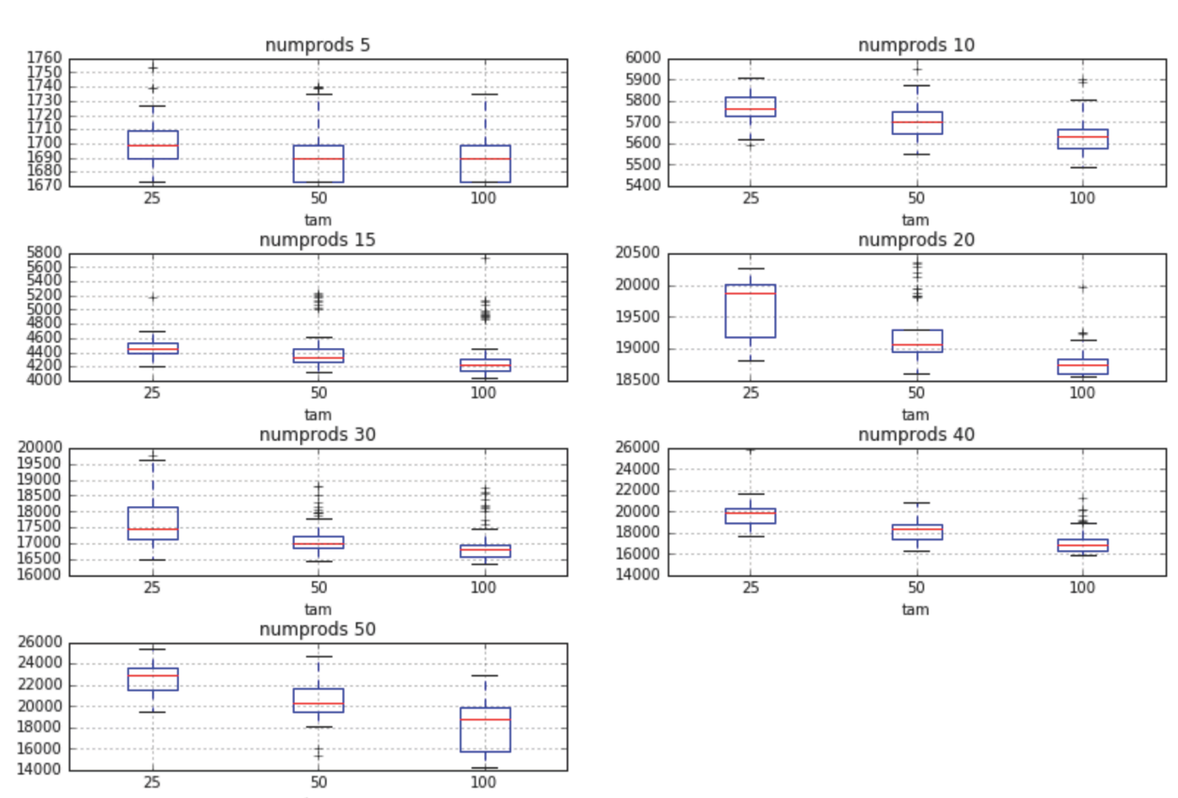
RESULTS

# of products	Exec. time	Float time	Delay
5	1.0000	-0.9999	0.8936
10	1.0000	-0.9998	0.9272
15	0.8897	-0.8702	0.8943
20	0.9973	-0.9972	0.9990
30	0.9907	-0.9837	0.9917
40	0.9907	-0.9945	0.9929
50	0.9956	-0.9831	0.9767

Correlation between fitness function and the execution time, float time and delay.

In order to study the performance of the proposed algorithm, seven synthetic production orders have been generated:

- Composed by sets from 5 to 50 jobs.
- 20% of an order's Jobs is dependent on other Jobs.
- Each job can be executed following an average of 3 routes.
- Each route has an average of 3 operations.
- Each operation requires a workforce between 1 and 10 operators.



Fitness value as a function of the population size for different number of products.

PARAMETERS

Each production order has been scheduled 10 times analyzing performance values for the following combination of parameters:

- Genetic population size: 25, 50, 100 (individuals)
- Death rate: 10%, 50%
- Elitism (selection): [0, 10] %
- Cross-over rate: 20%, 50%, 80%
- Mutation rate: [0,20]%
- Jobs increment: 5, non-incremental.
- Stagnation at 10000 epochs.

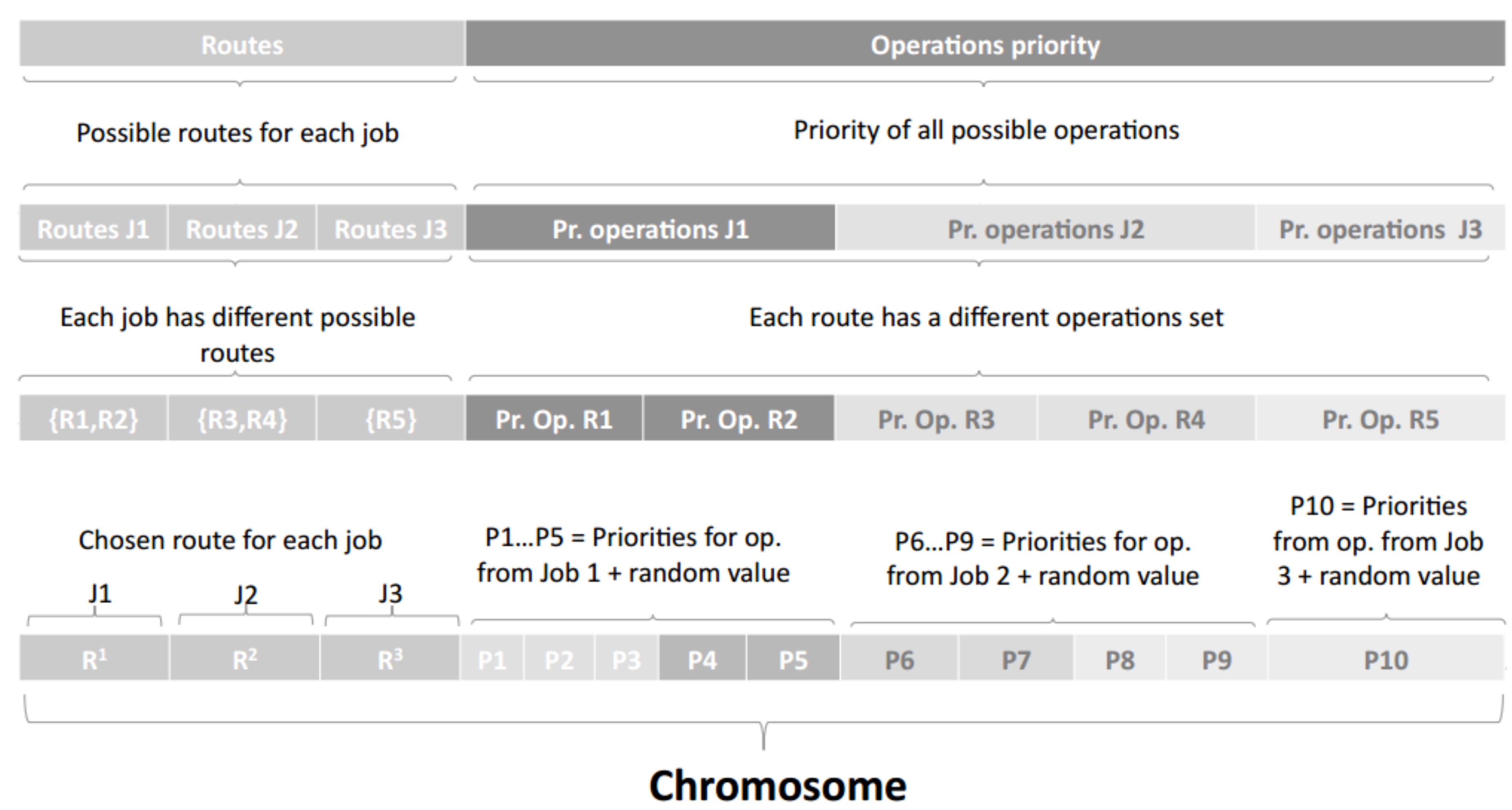
PROBLEM DESCRIPTION

A production order is composed by a set of jobs that must be executed using the different production lines of a factory.

Each job has a deadline and priority values defined. A set of constraints of different kinds must be satisfied:

Temporal Constraints: The time window in which the task should be completed.
Procedural Constraints: The precedence order in which each task must be executed.
Resource Constraints: Are enough resources available when they will be needed?

EXAMPLE



Task	Routes	Id	Operations
j_1	r^1_1	1	$O_1^{1,1}, O_2^{1,1}, O_3^{1,1}$
	r^1_2	2	$O_1^{1,2}, O_2^{1,2}$
j_2	r^2_1	3	$O_1^{2,1}, O_2^{2,1}$
	r^2_2	4	$O_1^{2,2}, O_2^{2,2}$
j_3	r^3_1	5	$O_1^{3,3}$

Example of possible routes that can be executed in a hypothetical planning

In this example the production order is composed by three tasks J_1, J_2 and J_3 . job J_1 can be implemented on two routes: $R^1=\{r^1_1, r^1_2\}$. job J_2 can be implemented on two different routes too $R^2=\{r^2_1, r^2_2\}$, however the task j_3 should follow just the route $R^3 = \{r^3_1\}$. The 'Operations' column shows the sequential order of operations stipulated by the route, necessary to execute the job.

CONCLUSIONS

A solution for a real world scheduling problem has been proposed using genetic algorithms with a priority encoding scheme. The main novelty in the proposal is the type of chromosome used in the GA that permits to define possible solutions to the problem in a very simple way, taking into account all specified problem variables and restrictions. A fitness function that includes a penalty term related to job production delays seems to be effective, based on the results obtained so far and on a correlation analysis performed. As an overall conclusion, the present proposal has permitted to find acceptable solutions in real time (two minutes time were allowed for obtaining a solution) and further improvements are underway, mainly by introducing an incremental approach that will permit the application of the present proposal to orders with more than 50 products.

ACKNOWLEDGMENTS

The authors acknowledge support through grants TIN2014-58516-C2-1-R and TIN2014-58516-C2-2-R from MINECO-SPAIN which include FEDER funds. LF and JMJA acknowledge support from Universidad de Málaga.