Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Scheduling (CI-Sched 2007)

Refinery Scheduling Optimization using Genetic Algorithms and Cooperative Coevolution

Leonardo M Simão Douglas M Dias and Marco Aur lio C Pacheco

Abstract — **Oil refineries are one of the most important examples of multiproduct continuous plants, that is, a continuous processing system that generates a number of products simultaneously. A refinery processes various crude oil types and produces a wide range of products. It is a complex optimization problem, mainly due to the number of different tasks involved and different objective criteria. In addition, some of the tasks have precedence constraints that require other tasks to be scheduled first. In this paper the refinery scheduling problem is addressed using genetic algorithms and cooperative coevolution. A simple refinery, with commonly found types of equipments, tasks and constraints of a real refinery, was created. Three test scenarios were designed with different sizes, demands and constraints. In all of them, the results obtained were far better than the ones obtained through random search.**

I Motivation

OIL refineries are one of the most important examples of multiproduct continuous plant in other words a continuous processing system which generates multiple products The optimization of the refinery scheduling is considered a complex problem due to the number and diversity of tasks and objectives Moreover it is a problem with precedence constraints: the scheduling of some tasks depends that other tasks were scheduled first

We can list a number of reasons why this problem should be solved and studied as follows:

- Daily costs in a refinery are very high and if they can be minimized it is worth the efforts;
- The scheduler needs to act quickly whenever an new and unexpected situation arises equipment maintenance for example) and the scheduling must be changed as quick as possible to avoid losses;
- Today there is no tool on the market that optimizes this problem and satisfies all the refinery scheduler needs;
- Scheduling problems are well known and very hard to solve in reasonable time;

Manuscript received January 15

Leonardo M Simão is with the Chemtech – A Siemens Company Rua da Quitanda 5 1º andar Centro Rio de Janeiro RJ Brasil 11 e mails: leonardo simao@{gmail com chemtech com br})

Douglas M Dias is with Pontif cia Universidade Católica do Rio de Janeiro Rua Marqu s de São Vicente 5 Gávea Rio de Janeiro RJ Brasil 45 9 e mail: douglasm@ele puc rio br)

Marco Aur lio C Pacheco is with Pontif cia Universidade Católica do Rio de Janeiro Rua Marqu s de São Vicente 5 Gávea Rio de Janeiro RJ Brasil 45 9 e mail: marco@ele puc rio br)

II Introduction

In general a refinery processes one or more type of crude oil producing a number of derived products as the LPG Liquefied Petroleum Gas) naphtha kerosene and the diesel oil [1]

We can separate the production control on a refinery into three levels: Planning Scheduling and Operation Figure 1 shows this structure

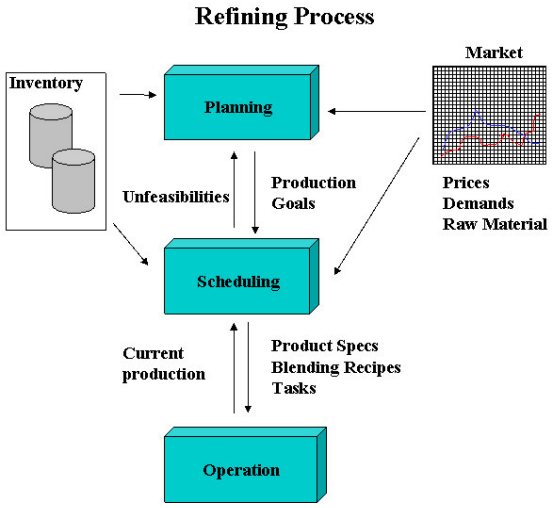

Fig 1 The Refining Process

The schedule defines how each product should be produced considering the following:

- A production preview defined by the planning products prices dates and quantities);
- Quantities and qualities of the raw material;
- Price quantity and quality of intermediary products actual inventory);
- Restriction on storages and pipeline paths topology of a plant);
- How the finished products are delivered

Everyday the production scheduler defines the tasks to be performed by each equipment during a time frame in order to make the best of the planned production In other words the final result of the schedule is a daily set of tasks to be executed in the refinery

The objective of this paper is to show the applicability and performance of the evolutionary computation in the

optimization of an oil refinery scheduling considering different refining stages In particular this work finds a system capable to deal with the problems as a whole considering every stage of refining and the objectives to be optimized

III Evolutionary Computation and Coevolution

The Evolutionary Computation is the optimization algorithm based on Darwinian s theory of the species evolution and in genetics The probabilistic algorithm gives a parallel and adaptive search mechanism based on the survival criteria of the most apt and on the reproduction of species The mechanism is obtained from a population of individuals solutions) represented by chromosomes binary words vectors matrixes etc) each one associated to an aptitude evaluation of the solution in the problem) which go through several cycles of an evolution process selection reproduction crossing and mutation)

The cooperative coevolution was inspired in the symbiosis concept where two or more species interact and collaborate with each other s evolution The evolution of each species happens in its niche or specialty Therefore their union can collaborate to the evolution of the ecosystem as a whole As in nature the species are genetically isolated; the individuals can only reproduce with other individuals of the same species which belongs to different populations The species only interact with one another through a shared domain and their relation is only of cooperation

The developed model to deal with refining optimization is presented in section V

IV Related Works

Scheduling problems have been investigated intensively in the areas of operations research and artificial intelligence Traditionally scheduling research has focused on methods for obtaining optimal solutions to simplified problems for example with integer programming or branch and bound algorithms In order to determine an optimal solution different restrictions were imposed on the problem domain for example on the number of jobs or machines which made the application of the results to more complex problems very difficult or even impossible

Due to the difficulty of the scheduling domain in many real world scheduling environments the objective is the determination of a feasible schedule in reasonable time which need not necessarily be an optimal schedule but should of course be as good as possible [ ]

Operations research methods for scheduling are described in more detail in [8]

V Coevolutionary Model for the Scheduling Optimization

The model suggested to solve this problem is a cooperative coevolutionary model [ ] composed by two species

To solve a problem using cooperative coevolution the most common approach is to decompose the problem into subcomponents Each subcomponent is then associated to a species or population The evolution of each species is independent except on evaluation As each individual of one species represents only part of the whole solution individuals from the other species must be selected using some criteria to allow the evaluation of the complete solution This is called collaboration Figure shows how this architecture works

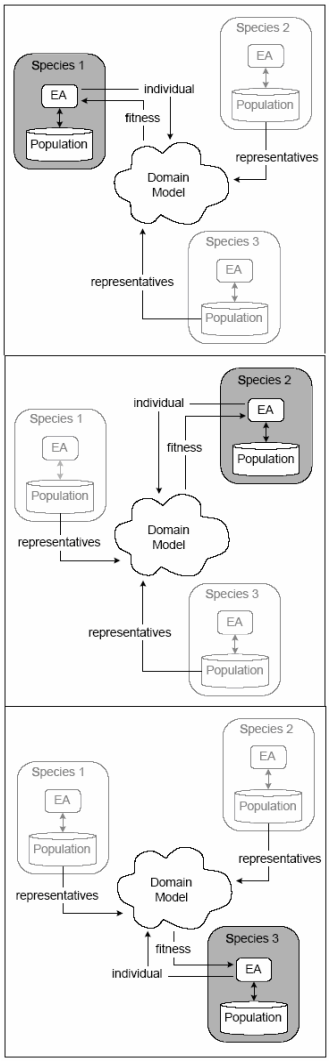

Fig Coevolutionary model of three species shown from the perspective of each in turn [ ]

In our model the first species will decide which task is to be planned whereas the second one will decide which resources will execute the task We decided to decompose the problem this way because when we represent the complete solution in one species these two aspects – *when to schedule a task* and *with which resource* – are often

conflicting

A. *Representation*

The representation chosen for this work deals with two species which have a parallel evolution of various aspects of the problem and collaborate for the solution of the problem

The first species is called *Allocation on Time* and deals with two structures: the priority list and the precedence graph:

- The priority list has the priority of each task
- The graph has the precedence relationships of the tasks i e which tasks should be schedule before the others

Priority List

Task	A	B	C	D	E	F
Priority	5	1	2	6	4	3

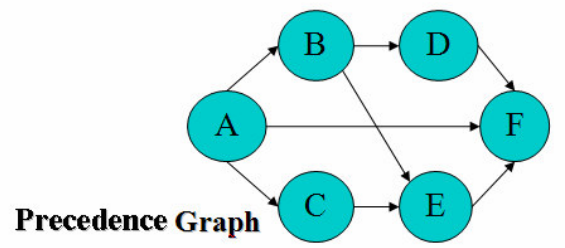

Fig The representation of the *Allocation on Time* species

Figure shows the two structures comprising the representation of the *Allocation on Time* species The priority list is the chromosome where the genetic operators will act Next in Figure 4 a fragment of the algorithm to select the tasks to be scheduled from the chromosome of species 1 is presented

It is important to notice that the algorithm shows how the system uses the priority list using the precedence graph to select the tasks to be scheduled respecting the precedence constraints

```
procedure Task scheduling
graph ← priority graph;
while (size(graph)>0)
    if every vertex has precedent then
       invalid graph;
       exit;
    else
        v ← vertex with higher priority
between the ones with no precedent;
        schedule v;
        remove v from the graph and
every connection that come from it;
    end if
end while
```

Fig 4 Algorithm to schedule tasks from species 1

The second species is called *Resource Allocation* and deals with a vector list Each position of a vector is associated to a task whereas each list has the resources that can execute the task The algorithm tries to allocate the first available resource following their position on the list Figure 5 shows this structure

Task	A	B	C	D	E	F
Resource	5	1	2	3	7	3
	1	4		4	2	5
	8			7		6
				9		

Fig 5 The representation of the *Resource Allocation* species

For example if the task "A" is *Distillation Unit Feeding* the tanks labeled 5 1 and 8 are the only ones that can feed the unit probably reflecting the topology of the plant) And in this chromosome fig 5) tank 5 will be selected to feed the unit unless it violates any other constraints

B. *Decoding*

The chromosome s decoding is performed by combining the information of each species Figure 4 below shows the essence of the algorithm

```
procedure Decode
begin
    chromossome1 ← Time individual selected;
    chromossome2 ← Resources individual
selected;
    while (there are tasks to schedule)
    begin
        select task using chromosome1;
        select resource for task using
chromosome2;
        schedule task;
    end while
end
```

Fig 6 Decoding Algorithm

Note that to decode the chromosome and to build a complete solution it is necessary to chose one individual of each species or collaborator as we mentioned earlier There are several methods for the selection of collaborators

Wiegand et al [ ] identifies three attributes when choosing a collaboration model:

- Collaborator selection pressure – The degree of greediness of choosing a collaborator;
- Collaboration pool size – The number of collaborators per subpopulation to use for a given fitness evaluation;
- Collaboration credit assignment – The method of assigning a fitness value given multiple collaboration driven objective function results;

And there are three options for credit assignment:

- Optimistic – The more traditional method of assigning an individual a fitness score equal to

the value of its best collaboration

- Hedge – Assign an individual a fitness score equal to the average value of its collaborations as is generally done in competitive coevolution
- Pessimistic – Assigning an individual a fitness score equal to the value of its worst collaboration

The results shown in this paper are using the optimistic method where the individual receives as its *fitness score* the evaluation of its best collaboration because they presented better results than the other ones in almost all experiments

C. Evaluation

The objective function must incorporate each scheduling objective:

- *Fulfill the demand for products.* The items for sale or shipping must be supplied with no delay;
- *Minimize the production costs.* Minimize raw material cost and minimize operational cost Raw material cost can be calculated from the quantities of certain oil that will be distilled during the schedule Operational costs on the other hand are related to how the plant operates avoiding frequent changes of process units operational modes and avoiding switch of tanks during an activity;
- *Assure products on spec.* The properties of commercialized derived products must be within the minimal and maximum limits specified;

The cost for failing to attend the demand can be quantified by a penalty delivery delay of finished products In order to make this cost proportional to the importance of the delay in relation to other deliveries we calculated the price based on the share of the delayed item:

$$C_{AD} = \sum_{i \in IV} product\ price_i \quad volumenotd\ elivered_i \qquad 1)$$

Where *IV* is the set of finished products to be delivered during the scenario

To calculate the cost of raw material we only need to multiply the price of crude oil spent on distillation by the quantity processed:

$$C_{MP} = \sum_{j \in UDA} \sum_{i \in C} price_i \quad processedvolume_{ij} \qquad)$$

Where *UDA* is the set of distillation units and *C* is the set of crude oil types used by the refinery

The operational costs considered in this work include only the number of storage tank or sphere) switches during a task transfer arrival of crude or deliver of product) We admitted that the number of *Unit Operation Mode changes is fixed by scenario* according to planning decisions and these changes are known *a priori* Therefore this cost can be described as:

$$C_{OP} = \sum_{i \in AT} storageswitches_i \quad switching\ cost \qquad)$$

Where *AT* is the set of transferring activities and the *switching cost* is arbitrary representing the cost of switching a tank during a transfer

The cost for having products out of specification is handled by removing their share on the revenues An *out of spec* product has no value Therefore we consider that all volume stored will be a cost on the scheduling

$$C_{DE} = \sum_{i \in IV} product\ price_i \quad outofspecvolume_i \qquad 4)$$

In the end the objective function FO) to be minimized where w_i represents the weight of each cost:

$$FO = w_1 C_{AD} + w\ C_{MP} + w\ C_{OP} + w_4 C_{DE} \qquad 5)$$

This objective function was never proposed by other researchers that we know of and is a direct way to compose all objectives It allowed us to test how much importance each cost has on the overall evolution in the case study scenarios

D. Operators

The operators applied to the individuals of both species are typically used in sequencing problems as the traveling salesman problem) It is important to notice that in the *Resource Allocation* species the operators act in the allele of the genes that means see Fig) in the list of resources of each activity but not among activities

We employed the following crossover operators: *Order Crossover* OX) *Partially Mapped Crossover* PMX) and *Cycle Crossover* CX) The ones used in mutations were *Swap* SM) and *Position Inversion* PI) They are presented and discussed in more detail by Michalewicz [4]

E. Time Representation

As exposed in [5] time is an important variable for scheduling problems For example if the scheduling time span is of seven days during this entire time frame the distillation unit must have been fed with oil at a certain flow How many different transfers from different tanks) to this process unit will take place?

Usually time slices with an equal duration are defined throughout the scheduling horizon as in the approach presented in [6] with number and duration of time slices arbitrarily defined Relevant decisions may happen in the boundary between two time slices It means that depending on the granularity of the problem a big number of time slices can be generated and the search for an optimal solution may become almost impossible

We decided not to split the time in fixed time slices but to split the quantities transferred by each task A parameter

Planning Information

Scenario 1

	1/1/03				2/1/03	
Demands	8h	12h	16h	20h	0h	4h
LPG -1	450					
LPG -2		450				
LPG -3			450			
LPG -4				450		
LPG -5					250	
Naphtha	1000	1000	1000	1000	1000	
Kerosene	1200	1200	1200	1200	1200	
Diesel 2-1			5000			
Diesel 2-2				2000		
Diesel 1					2000	

Crude Arrivals	8h	12h	16h	20h	0h	4h
-						

Operational Modes	8h	12h	16h	20h	0h	4h
Distillation Unit (UDA)	Normal					

Maintenance	8h	12h	16h	20h	0h	4h
-						

Scenario 2

Tempo	1/1/03				2/1/03	
Demands	8h	12h	16h	20h	0h	4h
LPG -1	450					
LPG -2		450				
LPG -3			450			
LPG -4				450		
LPG -5					250	
Naphtha	1000	1000	1000	1000	1000	
Kerosene	1200	1200	1200	1200	1200	
Diesel 2-1			5000			
Diesel 2-2				2000		
Diesel 1					2000	

Crude Arrivals	8h	12h	16h	20h	0h	4h
-						

Operational Modes	8h	12h	16h	20h	0h	4h
Distillation Unit (UDA)	Normal				Qcap	

Maintenance	8h	12h	16h	20h	0h	4h
TQ03						
TQ11						
TQ07						

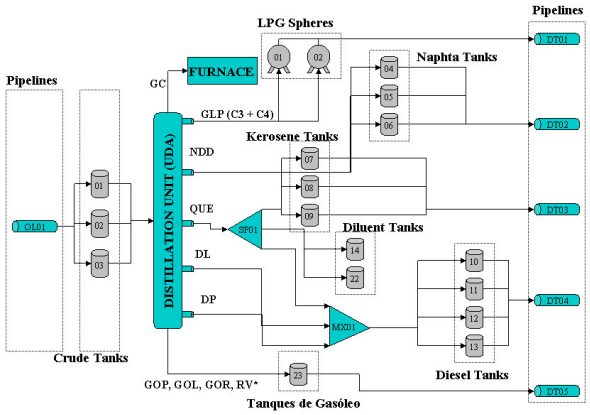

Fig Refinery used in the case study

TABLE I
REFINERY INITIAL STATE

Storage	Current Volume	Composition	Density g cm)	Sulphur m)
TQ 1	45	Crude 1	89	496
		Crude 1 Crude		
TQ	6	Crude 1	9 9	55
TQ		Crude 8 Crude	91	6695
TQ 4	155	Naphtha	1	15
TQ 5	65	Naphtha	1	15
TQ 6	1	Naphtha	1	15
TQ	151	Kerosene	8	1
TQ 8	151	Kerosene	8	1
TQ 9	11	Kerosene	8	1
TQ1	1	Diesel	88	45
TQ11	1	Diesel	8	1
TQ1		Diesel	86	4
TQ1	1	Diesel	89	5
TQ14	95	Diluent	85	1
TQ	95	Diluent	85	1
TQ	6	Diluent	9	5
EF 1	11	LPG	55	
EF	16	LPG	56	

Scenario 3

	1/1,03				2/1,03						3/1,03						4/1,03						5/1,03						6/1,03	
Demands	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h
LPG	250	250	250	250	250		500						500						500						500					
Naphtha	5000							3000						3000						3000						3000				
Kerosene	6000							2400						2400						2400						2400				
Diesel 2		5000	2000				4000		2400				4000		2400				4800						4800					
Diesel 1					2000																									

Crude Arrivals	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h
PET1							18000		18000																36000					
PET3													36000						36000											

Operational Modes	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h
Distillation Unit (UDA)	Normal						Qcap						Normal						Qcap						Normal					

Maintenance	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h	8h	12h	16h	20h	0h	4h
-																														

Fig 8 Planning information used in test scenarios

called *maximum transferred volume* was created to allow the algorithm to split the transfers of material This parameter decides the maximum volume for each transfer piece This solution allows the user to choose task size according to the type of the task

VI CASE STUDY

A. *Prototype Refinery*

The refinery plant we chose for this work is a simple plant presented in [6] which represents all types of equipment and constraints present in a real life refinery Figure)

Scenarios with different planning information were studied such as process unit operational mode changes and maintenance scheduling The number of days in the scenario scheduling horizon) has also varied to show the behavior of the algorithm due to the increasing number of activities

The refinery initial state content of the refinery tanks and spheres) is shown on Table I

In this plant two operational modes for the distillation unit were considered The objective of the first one *Normal*) is the production of aviation kerosene The second one *Qcap*) is used to produce asphalt diluents For each distillation unit operational mode the outlet streams for each crude oil fed volumetric fractions and property values for each output product) were defined

In figure 8 we can see the scenarios created for the evaluation of the coevolutionary algorithm performance The longer scenario has a total duration of five days and is able to evaluate the algorithm in a schedule that has a number of activities similar to the number of activities present in a real life situation The scenarios were defined according to examples from [6]

VII RESULTS

A. *Performance Analysis*

We made ten experiments with each scenario The performance charts are the average results for scenario

It is important to mention that the *random search* here uses basically the same structures as the *genetic algorithm* except the genetic operators So it does not generate unfeasible solutions For example it uses the precedence graph and the scheduling algorithms mentioned above

To demonstrate how difficult this problem is the random search in ten experiments with 6 individuals each with 5 6 s or 1h 4min) was never able to find a solution that would deliver finished products in spec without delay

Next the performance charts: objective function Figure 9) and separated objectives Figures 1 to 1)

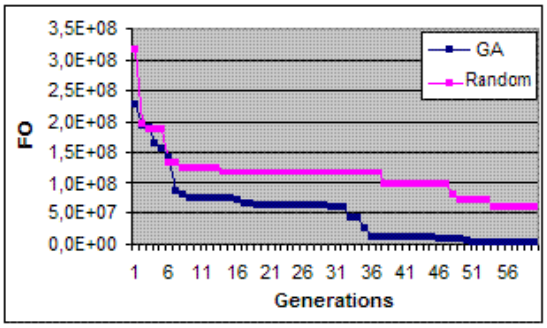

Fig 9 Performance of the genetic algorithm versus random search

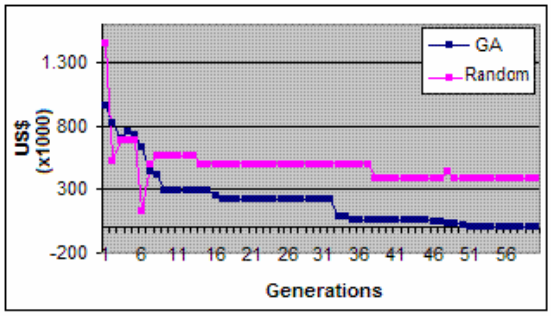

Fig 1 Cost of delivering delays

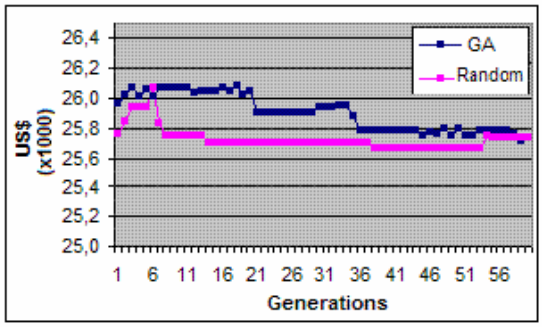

Fig 11 Raw material Crude) cost

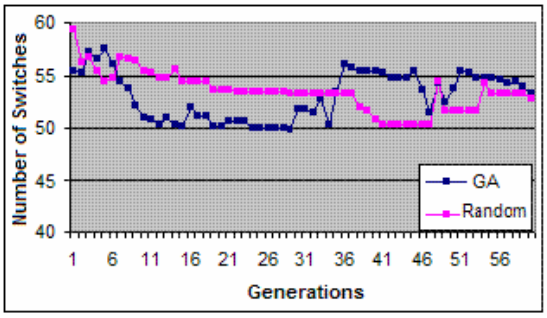

Fig 1 Number of storage switches during transfers

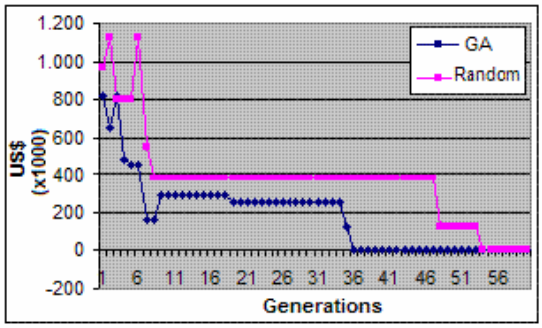

Fig 1 Cost for delivering out of spec products

As each one of the aspects is separately examined it seems that the random search is working as good as or even better than the evolutionary algorithm However as each one of the objectives is combined it becomes clear that the solutions achieved by the random search are inferior to the ones presented by the genetic algorithm Fig 9) The evolution of the objectives in the random search has a *greedy* behavior by trying to meet the objectives that are at the same

time easy with the minimum combination possible) and with a huge wide impact in the objective function In the genetic algorithm however the convergence occurs in a different way more gradually taking into consideration every objective: first by making a more global search in the space of the solutions and less greedy for then when close to the best solutions make a local search in more subtle objectives like the storage switching minimization

By observing these charts we can see that the genetic algorithm in average had a much more superior performance than the random search It is important to mention that even in greater scenarios than these the processing time does not make the use of algorithm impracticable as in most of the approaches for scheduling problems of the real world

Table II shows the results of one of the best solutions found in 6 generations of 1 individuals for the coevolutionary genetic algorithm for each scenario

TABLE II
PERFORMANCE OF ONE OF THE BEST SOLUTIONS FOUND

Parameter	Scenario 1	Scenario	Scenario
Generations	6	6	6
Population	1	1	1
Max Vol – Crude m)	6	6	6
Max Vol – Feed m)	6	6	1
Number of generated tasks	8	86	8
Time s)	1	18	146
Non delivered prods US)			
Raw material cost US)	4 88 4	4 88 4	5 156 5 89
Nr of Storage Switches	4	6	
Out of spec US)			
Evaluation	49 4	49 4	54 6 5 9

1/1/2003 08:00:00 - 2/1/2003 08:00:00 -> Operation Mode 'Normal (01/01/03 - 02/01/03)' on the 'UDA' Process Unit.
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Feeding Unit 'UDA' from storage 'TQ01' at a flow rate of 1500 m3/h (Volume = 6000 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Delivering Item 'GLP-1' from storage 'EF02' to pipeline 'DT01' at a flow rate of 112.5 m3/h (Volume = 450 m3).
1/1/2003 08:00:00 - 2/1/2003 04:00:00 -> Delivering Item 'QAV' from storage 'TQ09' to pipeline 'DT03' at a flow rate of 300 m3/h (Volume = 6000 m3).
1/1/2003 08:00:00 - 2/1/2003 04:00:00 -> Delivering Item 'NPTQ' from storage 'TQ05' to pipeline 'DT02' at a flow rate of 250 m3/h (Volume = 5000 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Gas Processing) from 'UDA' to 'CAFOR' at a flow rate of 1.8 m3/h (Volume = 7.2 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Store LPG) from 'UDA' to 'EF01' at a flow rate of 39 m3/h (Volume = 156 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Store Naptha) from 'UDA' to 'TQ04' at a flow rate of 207 m3/h (Volume = 828 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Store Gasoil) from 'UDA' to 'TQ23' at a flow rate of 766.2 m3/h (Volume = 3064.8 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Split Kerosene) from 'UDA' to 'SP01' at a flow rate of 174 m3/h (Volume = 696 m3).
1/1/2003 08:00:00 - 1/1/2003 12:00:00 -> Sending (Store Kerosene) from 'SP01' to 'TQ07' at a flow rate of 174 m3/h (Volume = 696 m3).

Fig 14 Fragment of a generated task list

The *maximum volume* parameter can be changed to vary the number of activities that will be generated by the algorithm thus affecting overall performance It s because in this representation the number of genes of the chromosome of both species is dependent on the number of activities For example setting the *maximum volume* to 1 m *volume slice*) for process unit feeding create tasks with duration of eight hours feeding and transferring outlet streams) which is not usually too much for a daily work on a refinery Then there is a chance to reduce the process time for longer scenarios

B. *Schedule Generation*

A fragment of a schedule generated by the system can be seen on Figure 14 It corresponds roughly to the task list used by the operation personnel at the refinery

VIII CONCLUSIONS

The objective of this work was to show the applicability of a cooperative coevolutionary model in the global optimization of refinery scheduling

A cooperative coevolutionary model was adopted to analyze the problem and to favor the performance of the system The decoder implemented to generate a feasible schedule from a chromosome used two species to schedule the tasks in time and with a specific resource always respecting precedence constraints solved by the graph of the first species) and the other operational plant constraints such as: maintenance preparation time of products among others By using this approach there is need to penalize correct or get rid of invalid individuals which makes the evolution much more efficient The evaluation function was conceived by incorporating several important objectives of the refinery scheduling problem

The cases we studied were represented by a refinery with all types of activities and constraints found on the daily operation of a real life refinery We validated and evaluated the performance of the model developed using these cases

The results proved the efficiency of the coevolutionary algorithm if compared to the random search or even if evaluated as high quality feasible solutions generated in acceptable time

The impacts of the number of generated activities on the

performance of the algorithm were evaluated too and it shows that we need to implement a better strategy to define the size of the time slices

Finally this work shows the application of evolutionary computation on the optimization of the refinery scheduling as a possible alternative practical and efficient in the solution of this challenging issue

IX FUTURE WORK

There is a lot of work yet to be done in this topic This is but the start of a promising approach to scheduling optimization We can list some improvements to be made in this model:

- Use a multi objective optimization technique to comprise all the objectives of the problem in a more adequate fashion;
- Application of the model in a real life refinery scheduling;
- Usage of more properties like: octane number viscosity etc ;
- Take into account the crude oil blending in the pipeline when arriving when two different crudes arrive in sequence blending occurs to some extent);
- Allow product transfer between tanks when really needed This may need some heuristics;
- Allow storage to send and receive at the same time which can be seen in some refineries;
- Allow two or more storages to feed a process unit at the same time blending product in the pipelines This is used in many refineries;
- Try to evolve other decision parameters that were fixed in this model Maybe as a new species;
- Change the time representation as seen in section V – topic E) to a more flexible solution instead of fixed time slices in order to allow a reduction in the number of scheduled tasks

REFERENCES

[1] A C Hax and D Candea *Production and inventory management* Prentice Hall Englewood Cliffs EUA 1984

[ ] M A Potter and K A DeJong "Cooperative coevolution: An architecture for evolving coadapted subcomponents", *Evol. Computation* vol 8 no 1 pp 1 9 Spring

[ ] R P Wiegand W C Liles and K A De Jong "An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms" *Proceedings of the Genetic and Evolutionary Conference* Morgan Kaufmann Publishers 1

[4] Z Michalewicz *Genetic Algorithms + Data Structures = Evolution Programs* nd Edition Springer Verlag New York USA 1994

[5] L F L Moro "T cnicas de Otimiza ão Mista Inteira para o Planejamento e Programa ão de Produ ão em Refinarias de Petróleo" PhD thesis Escola Polit cnica da Universidade de São Paulo São Paulo Brasil [in Portuguese]

[6] P Smania and J M Pinto "Mixed Integer Nonlinear Programming Techniques for the Short Term Scheduling of Oil Refineries" *Proceedings of the 8th International Symposium on Process Systems Engineering,* China

[ ] T Bäck D B Fogel and Z Michalewicz *Handbook of Evolutionary Computation* IOP Publishing Ltd and Oxford University Press 199

[8] J Blazewicz K H Ecker G Schmidt and J Weglarz *Scheduling in Computer and Manufacturing Systems* nd ed Springer Berlin 1994