



Proyecto Práctica Electiva: Parte final



Estudiante: Joaquín Fernández.
Memorista: Nicolás Claderón.
Profesor: Victor Reyes.



Contenidos de esta presentación

1. Documentación y estado del Arte.

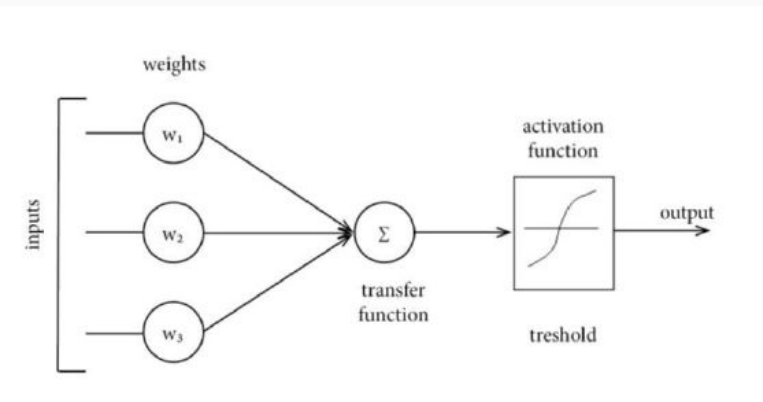
- a. Anteproyecto Nicolás Calderón.
- b. Interval Branch-and-Bound algorithms for optimization and constraint satisfaction: a survey and prospects by Ignacio Araya & Victor Reyes.
- c. The Simplex-simulated Annealing Approach To Continuous Non-Linear Optimization by Margarida F. Cardoso, R. L. Salcedot and S. Feyer De Acevedo.
- d. Predicting the Execution Time of the Primal and Dual Simplex Algorithms Using Artificial Neural Networks.
- e. Nonlinear Models and Problems in Applied Sciences from Differential Quadrature to Generalized Collocation Methods

2. Profundizando la herramienta Ibex (Contexto de instalación y uso general).

3. Metodología propuesta.

- a. Esquema de solución (pseudo-algoritmo)
- b. Arquitectura de la aplicación.
- c. Criterios aplicados
- d. Análisis y resultados.
 - i. Gráficas en 2D.
 - ii. Matriz de Confusión

4. Conclusiones y acotaciones.

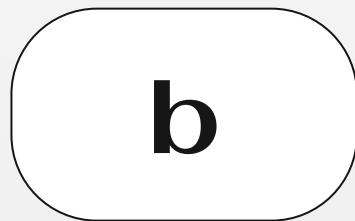
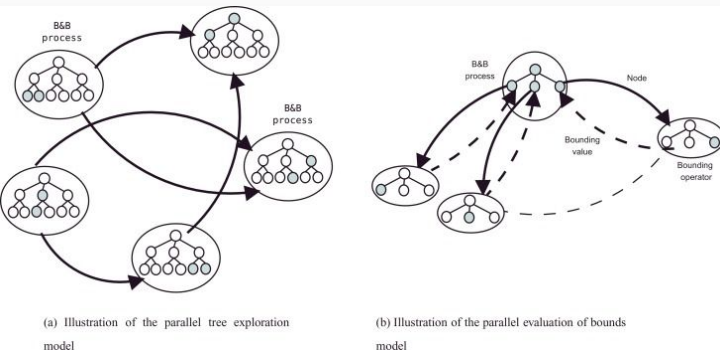


a

Anteproyecto Nicolás Calderón

- Branch and Bound (B & B).
- Modelos No Lineales.
- Dominios Continuos.
- Simplex.
- MLP.

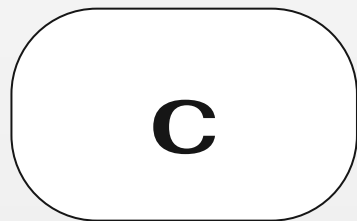
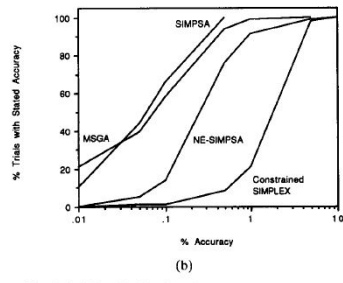
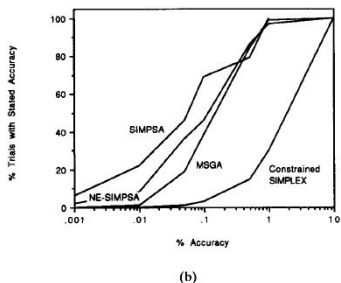
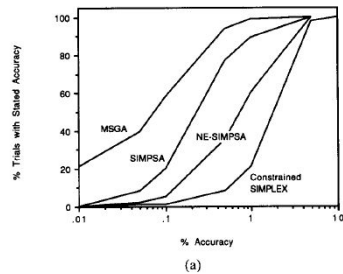
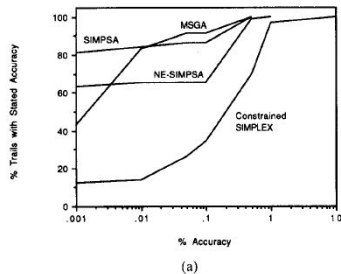




Interval Branch-and-Bound algorithms for optimization and constraint satisfaction: a survey and prospects por Ignacio Araya y Victor Reyes

- Branch and Bound (B & B).
- Contractors.
- Bisectores.
- CP.





The Simplex-simulated Annealing Approach To Continous Non-Linear Optimization by Margarida F. Cardoso, R. L. Salcedot and S. Fayo De Acevedo (1994).

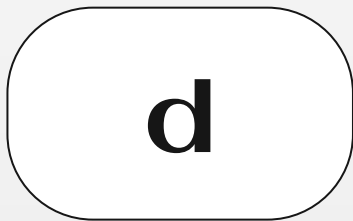
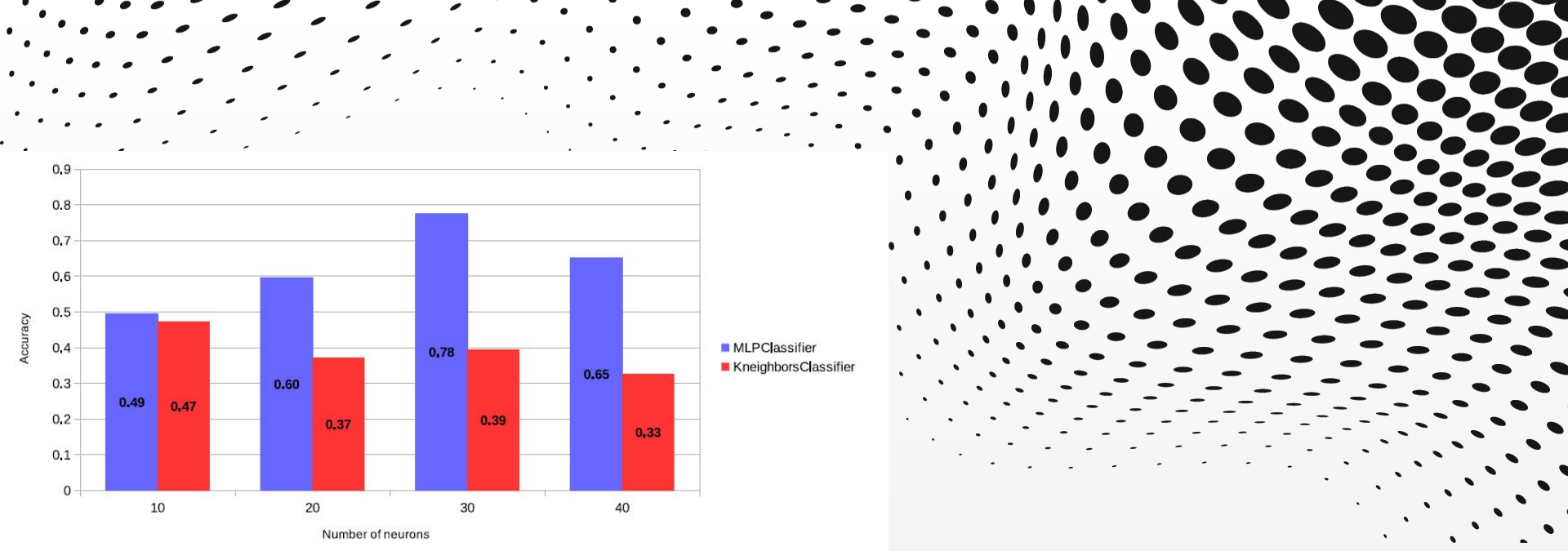
- Press y Teukolski (1991).
- SIMPSA.
- NE-SIMPSA.
- MSGA.



Comparativa de resultados

Function	Starting point	Simplex	SIMPSA	NE-SIMPSA
7	$(25, 1, 1)^T$	845	44413	23839
	$(30, 0, 0.6)^T$	631	44622	24651
8	$(1, 1, 1, 0.3, 0)^T$	954	59842	23970
	$(0, 0, 0.3, 1.2, 0)^T$	1546	46734	23214
9	$(28, 14, 14, 110, 110, 110, 110)^T$	2040	52402	13780
			298228*	60037*

* $\delta = 10^{-2}$ for SIMPSA and $\delta = 10^{-5}$ for NE-SIMPSA.



Predicting the Execution Time of the Primal and Dual Simplex Algorithms Using Artificial Neural Networks

- Simplex Primal y Simplex Dual.
- Algoritmos de Regresión y Clasificación.
 - 4 clases para el tiempo de ejecución.
- Desplazamiento en politopo.



Comparativa de resultados

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

Table 7. Classes of the primal and dual simplex algorithm execution time (in seconds).

Class	Primal	Dual
0	$0 < \text{time} < 0.1$	$0 < \text{time} < 0.1$
1	$0.1 \leq \text{time} < 0.5$	$0.1 \leq \text{time} < 1$
2	$0.5 \leq \text{time} < 4$	$1 \leq \text{time} < 10$
3	$4 \leq \text{time}$	$10 \leq \text{time}$

Comparativa de resultados

Table 10. Classification reports for the primal and dual simplex algorithm execution time.

Class (Primal)	Precision	Recall	F_1	Support
0	0.92	0.92	0.92	24
1	0.91	0.88	0.89	24
2	0.74	0.87	0.80	23
3	0.73	0.61	0.67	18
avg/total	0.83	0.83	0.83	89

Class (Dual)	Precision	Recall	F_1	Support
0	0.89	1.00	0.94	33
1	0.80	0.83	0.82	24
2	0.88	0.71	0.79	21
3	0.70	0.64	0.67	11
avg/total	0.84	0.84	0.84	89

02



Profundizando Ibex y reuniones con profesor y memorista



<http://www.ibex-lib.org>

ibex-team/ibex-lib

IBEX is a C++ library for constraint processing over real numbers.



23

Contributors

44

Issues

9

Discussions

58

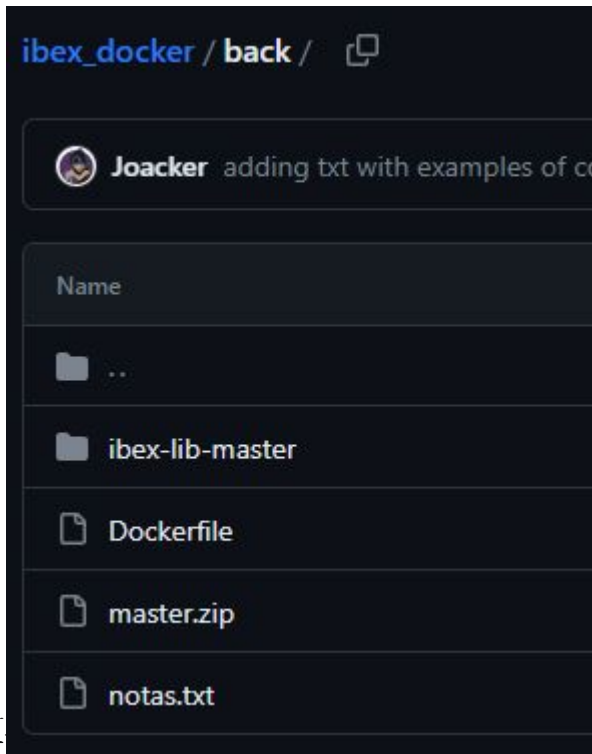
Stars

47

Forks



Herramienta aplicada en contenedor



```
1 FROM ubuntu:22.04
2
3
4 RUN apt-get update && apt-get install -y \
5     bash \
6     g++ \
7     gcc \
8     flex \
9     bison \
10    python2-dev \
11    make \
12    git \
13    build-essential \
14    cmake
15
16 RUN apt install -y \
17     python-is-python3
18
19 RUN apt install -y \
20     python3-pip
21
22 RUN apt install libeigen3-dev -y
23
24 RUN apt update
25 RUN apt upgrade -y
26
27 # Continue with your Dockerfile instructions
28 WORKDIR /app
29 COPY ./ibex-lib-master /app
30
31 RUN pip install -r requirements.txt
32
33 RUN chmod +x waf
34
35 # RUN ./waf configure --interval-lib=gao1 --lp-lib=soplex \
36 # && ./waf install
37
38 # ./_build_/src/ibexopt benches/optim/medium/alkylbis.bch --random-seed=1
39 # ./_build_/src/ibexopt benches/optim/hard/chembis.bch --random-seed=1 >> commands.log
40 # ./waf distclean
41
42 CMD ["tail", "-f", "/dev/null"]
```

Herramienta aplicada en contenedor

```
1  version: '3.7'
2
3  services:
4
5    zookeeper:
6      image: "bitnami/zookeeper:latest"
7      restart: always
8      environment:
9        - ALLOW_ANONYMOUS_LOGIN=yes
10     ports:
11       - "2181:2181"
12       - "2888:2888"
13       - "3888:3888"
14
15     kafka:
16       image: "bitnami/kafka:latest"
17       restart: always
18       ports:
19         - "9092:9092"
20       environment:
21         - KAFKA_CFG_ZOOKEEPER_CONNECT=zookeeper:2181
22         - ALLOW_PLAINTEXT_LISTENER=yes
23       depends_on:
24         - zookeeper
25
```

```
26  app:
27    image: app
28    build: ./back
29    container_name: app
30    ports:
31      - 81:8000
32    #make a volumes for an entire folder of cpp files
33    # volumes:
34    #   - ./back:/app&ibex-lib/examples
35    #make still runing container
36    stdin_open: true
37    tty: true
38    #make a container restart when it stops
39    restart: always
40    #make a container run in background
41    volumes:
42      - ./back/ibex-lib-master:/app
43      - ./results:/app/results
44    command: /bin/bash -c "while true; do sleep 1; done"
45
46  data:
47    image: data
48    build: ./data
49    container_name: data
50    ports:
51      - 82:8000
52    volumes:
53      - ./data:/app
54      - ./results:/app/results
55      - ./models:/app/models
56      - ./back/ibex-lib-master/benchs:/app/benchmarks
57
58
```

Herramienta aplicada en contenedor

PROBLEMS5OUTPUTDEBUG CONSOLETERMINALSQL CONSOLECOMMENTS

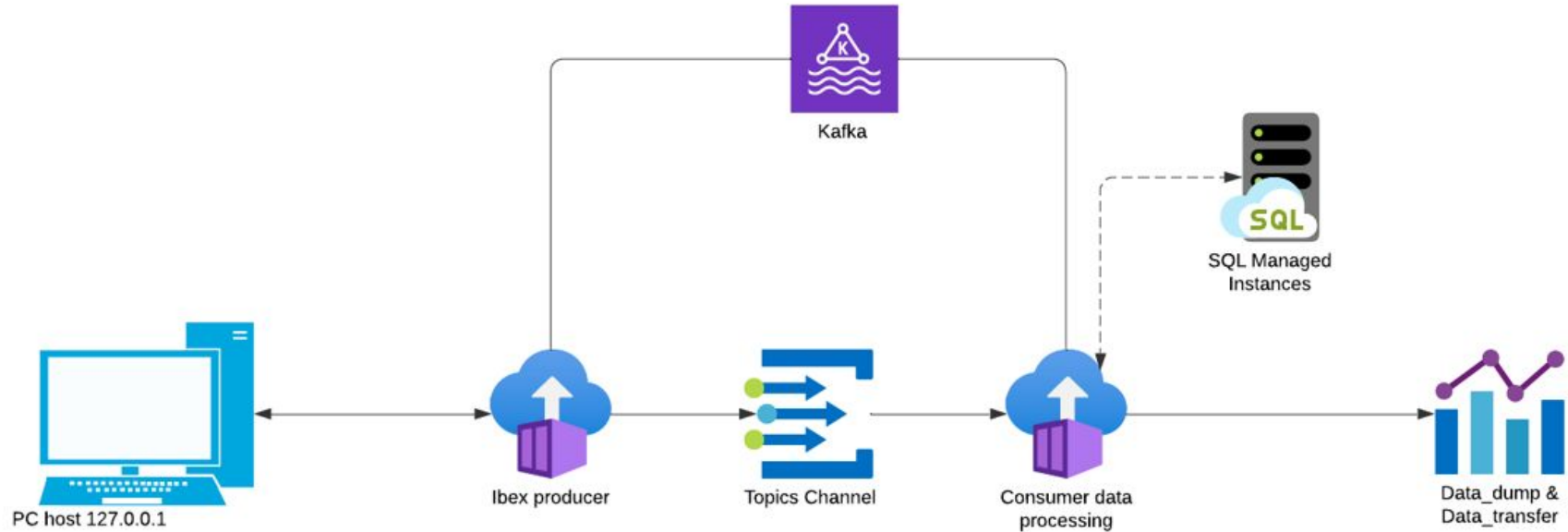
```
ibex_docker-kafka-1 | [2023-07-07 04:13:05,900] INFO [BrokerServer id=1] Finished waiting
for the broker to be unfenced (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,900] INFO [SocketServer listenerType=BROKER, n
deId=1] Enabling request processing. (kafka.network.SocketServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,901] INFO Awaiting socket connections on 0.0.0.
0:9092. (kafka.network.DataPlaneAcceptor)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,902] INFO [BrokerServer id=1] Waiting for all o
f the authorizer futures to be completed (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,903] INFO [BrokerServer id=1] Finished waiting
for all of the authorizer futures to be completed (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,903] INFO [BrokerServer id=1] Waiting for all o
f the SocketServer Acceptors to be started (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,903] INFO [BrokerServer id=1] Finished waiting
for all of the SocketServer Acceptors to be started (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,903] INFO [BrokerServer id=1] Transition from S
TARTING to STARTED (kafka.server.BrokerServer)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,904] INFO Kafka version: 3.5.0 (org.apache.kafk
a.common.utils.AppInfoParser)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,904] INFO Kafka commitId: c97b88d5db4de28d (org
.apache.kafka.common.utils.AppInfoParser)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,904] INFO Kafka startTimeMs: 1688703185903 (org
.apache.kafka.common.utils.AppInfoParser)
ibex_docker-kafka-1 | [2023-07-07 04:13:05,906] INFO [KafkaRaftServer nodeId=1] Kafka Serv
er started (kafka.server.KafkaRaftServer)
```

```
PS E:\Archivos del programa\laburar\practica_electiva\estudio\testing2\ibex_docker> docker
PS -a
CONTAINER ID   IMAGE                                COMMAND                                  CREATED        STATUS        NAM
PORTS
ES
724bf7a0cda4   bitnami/kafka:latest               "/opt/bitnami/script..." 16 seconds ago Up 14 sec
ibe
x_docker-kafka-1
9f5383014672   data                                "tail -f /dev/null"        16 seconds ago Up 14 sec
dat
onds 0.0.0.0:82->8000/tcp
a
4de9fc38aa43   app                                "/bin/bash -c 'while..." 16 seconds ago Up 14 sec
app
onds 0.0.0.0:81->8000/tcp
e679fa69a500   bitnami/zookeeper:latest          "/opt/bitnami/script..." 16 seconds ago Up 14 sec
ibe
onds 0.0.0.0:2181->2181/tcp, 0.0.0.0:2888->2888/tcp, 0.0.0.0:3888->3888/tcp, 8080/tcp
x_docker-zookeeper-1
PS E:\Archivos del programa\laburar\practica_electiva\estudio\testing2\ibex_docker>
```

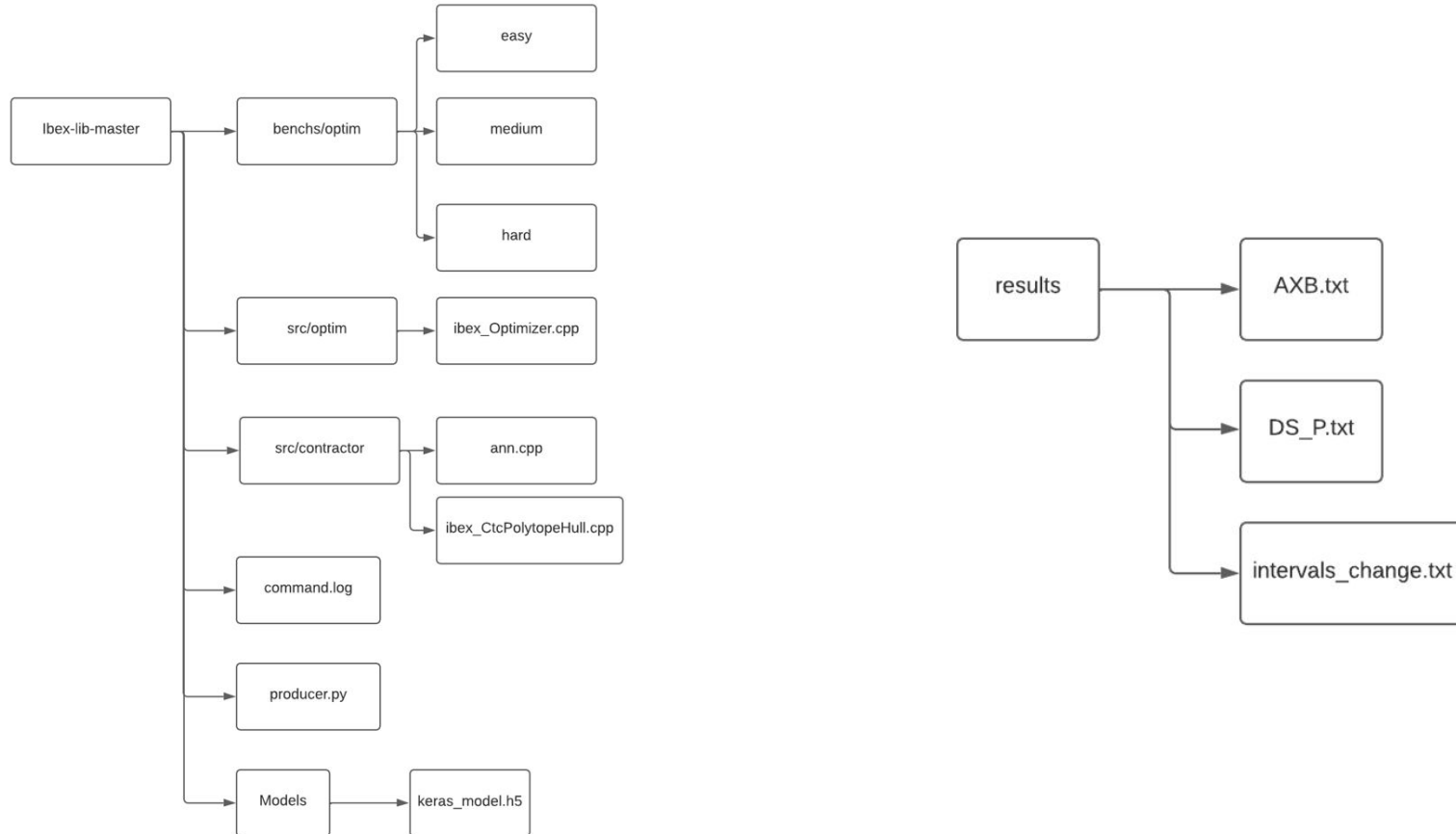
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live



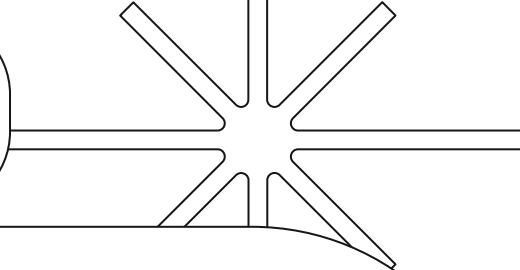
Ambiente de desarrollo y de pruebas



Arquitectura de ibex y en que se prestará atención



03



Metodología



a. Pseudo-algoritmo propuesto

1) Recopilación
Preparación
de datos

2) Diseño y
entrenamiento
de la red
neural

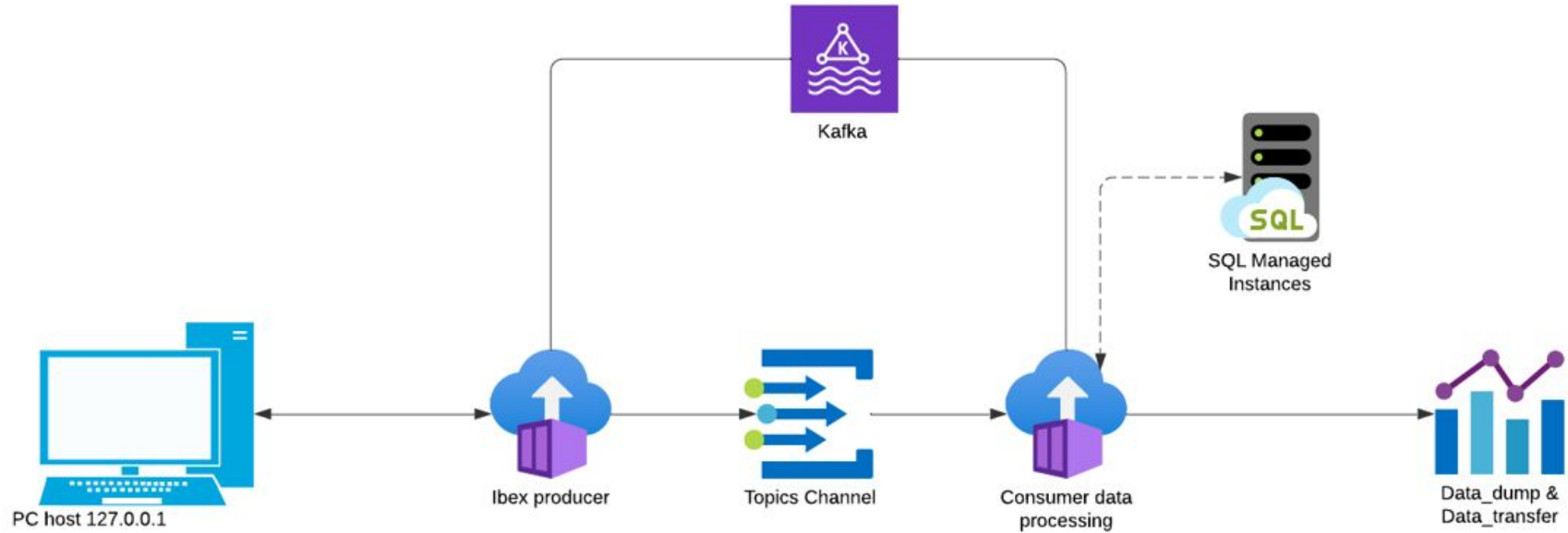
3) Validación
cruzada y ajuste
de
hiperparámetros

4) Evaluación
del rendimiento

5) Optimización
de los vértices
del politopo

6) Resultados y
conclusiones

B. Arquitectura de la aplicación



C. Criterios aplicados

- Variar probabilidad con respecto a una distribución srand

```
void CtcPolytopeHull::optimizer(IntervalVector& box, float proba) {  
  
    Interval opt(0.0);  
    int* inf_bound = new int[nb_var]; // indicator inf_bound = 1 means the inf bound is feasible or already contracted, call to simplex useless (cf Baharev)  
    int* sup_bound = new int[nb_var]; // indicator sup_bound = 1 means the sup bound is feasible or already contracted, call to simplex useless  
  
    for (int i=0; i<nb_var; i++) {  
        if (contracted_vars[i]) {  
            inf_bound[i]=0;  
            sup_bound[i]=0;  
        } else {  
            inf_bound[i]=1;  
            sup_bound[i]=1;  
        }  
    }  
  
    int nexti=-1; // the next variable to be contracted  
    int infnexti=0; // the bound to be contracted contract infnexti=0 for the lower bound, infnexti=1 for the upper bound  
    LPSolver::Status stat=LPSolver::Status::Unknown;  
  
    // Update the bounds the variables  
    mylinearsolver.set_bounds(box);  
    //make random to choice probas with srand  
    float probas[2]={proba,1-proba};  
    if (proba >= probas[0]){  
        for(int ii=0; ii<(2*nb_var); ii++) { // at most 2*n calls
```

C. Criterios aplicados

- Considerar la cantidad de variables de entrada para el cálculo de la precisión

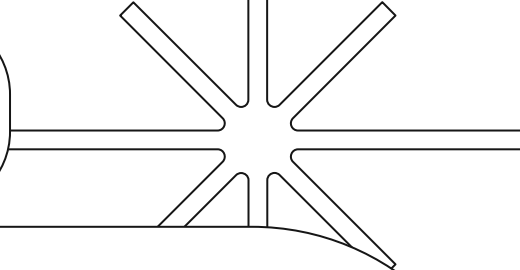
```
1 variables
2
3 x1 in [-32,32];
4 x2 in [-32,32];
5 x3 in [-32,32];
6 x4 in [-32,32];
7 x5 in [-32,32];
8
9 minimize
10
11 -20 * exp (-0.2 * sqrt (0.2 * (x1^2 + x2 ^2 + x3^2 + x4^2 + x5 ^2))) - exp (0.2 * (cos (2*pi*x1) + cos (2*pi*x2) +cos (2*pi*x3)
12 + cos (2*pi*x4) + cos (2*pi*x5))) + 20 + 2.718281;
13
14 constraints
15 x1 <= 500;
16 end
17
```

C. Criterios aplicados

- Timeout tras el paso de 45 (2700 s) minutos o 1 hora (3600 s)

```
clock_t t1,t2;
t1=clock();
for (int i=0;i<lr2->L_b.size();i++){
    AXB_for_red.push_back(lr2->L_b[i]);
    // stop post time 2700 s pass
    if (i==2700){
        t2=clock();
        float diff ((float)t2-(float)t1);
        float seconds = diff / CLOCKS_PER_SEC;
        if (seconds>2700){
            break;
        }
    }
}
```

04

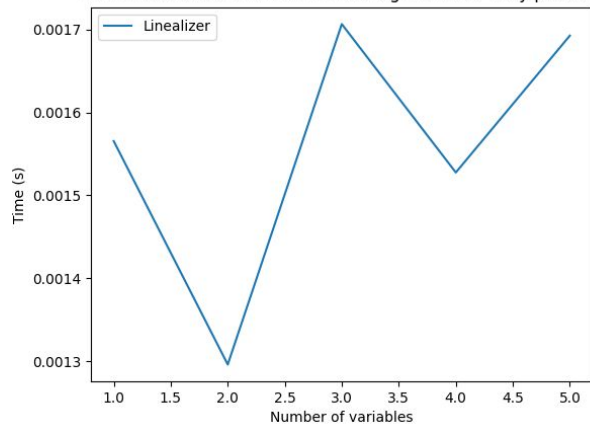


Resultados

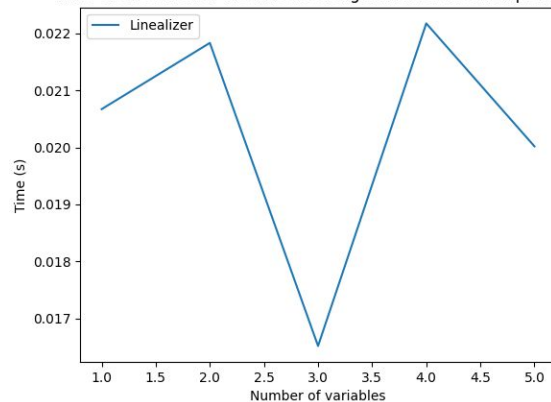


Graficas linealizer: Problemas easy, medium y hard

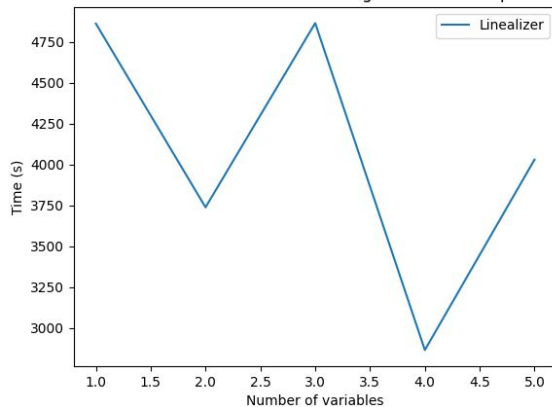
Time of execution of the Linealizer algorithm for easy problem



Time of execution of the Linealizer algorithm for medium problem

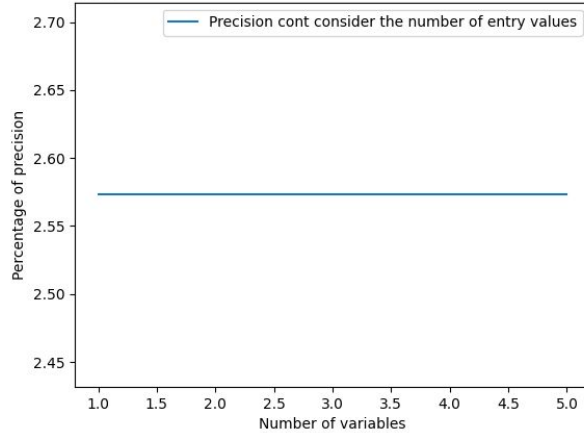


Time of execution of the Linealizer algorithm for hard problem

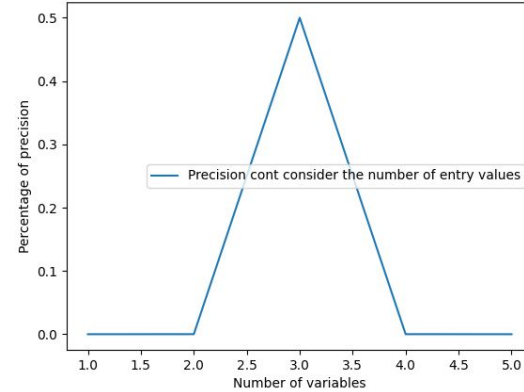


Gráficas Precisión: Problemas easy, medium y hard

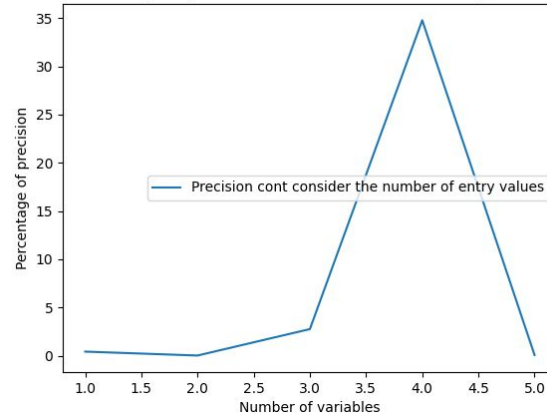
Percentage of precision of the iterations for easy problem
1e-8



Percentage of precision of the iterations for medium problem

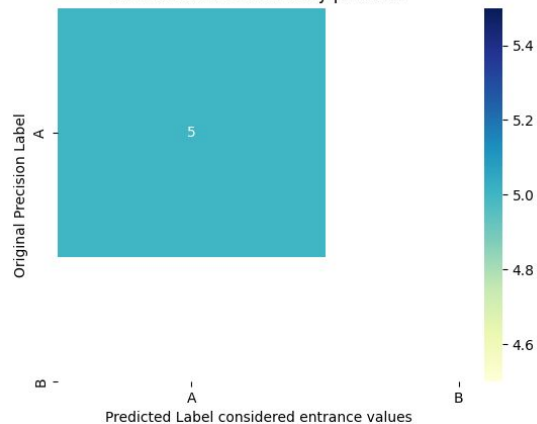


Percentage of precision of the iterations for hard problem

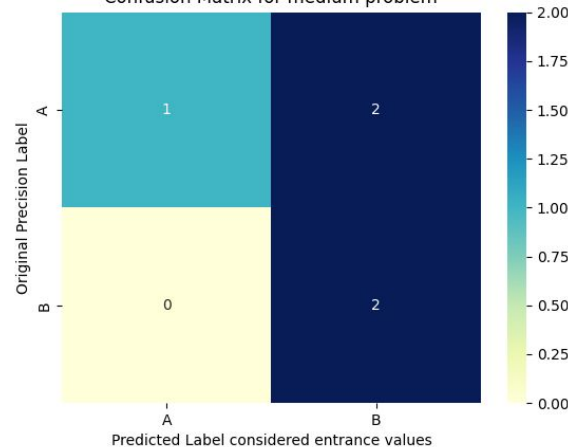


Matriz de Confusion: Problemas easy, medium y hard

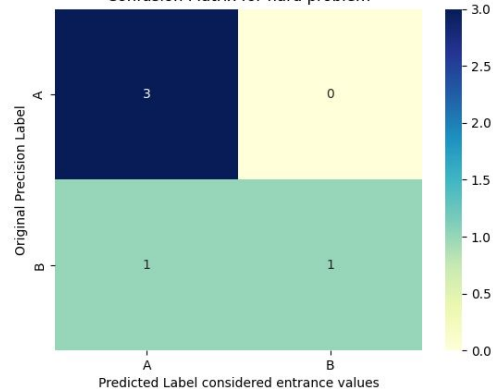
Confusion Matrix for easy problem



Confusion Matrix for medium problem



Confusion Matrix for hard problem





4. Conclusiones y acotaciones


1. Como posible solución podría verse el tema de la parametrización para tener control del flujo y estructura.
2. Probar nuevos modelos de redes neuronales ubicadas en keras.
3. Establecer mejores criterios para el armado de preconditionadores.
4. Evaluar usar más problemas dentro de ibex para establecer probabilidades o coeficientes de aprendizaje más apropiados.



Enlaces de videos de demostración

Parte 1: <https://www.youtube.com/watch?v=sgeV153Rnaw>

Parte 2: <https://www.youtube.com/watch?v=nC12ZCvvi-g>



Enlaces de repositorios

https://github.com/Joacker/ibex_docker

https://github.com/Joacker/Preconditioners_Simplex

https://github.com/hohenheim99/Preconditioners_Simplex

<https://github.com/moof2k/kerasify>

GRACIAS TOTALES

¿Alguna duda? Pregunta?
Consulta? Incertidumbre?
Vacilación? Interrogante?
Dubitación ? Disyuntiva?

