

Comparación de algoritmos meta-heurísticos para la programación de tareas en la nube

Torres Lima Ruben Jesus
Valencia Rivero Carlos David

Introducción

Scheduling : Aborda el problema de cuántos recursos se necesita asignar a una tarea específica. Lo que mejora la calidad del servicio (dependiendo del servicio).

Task scheduling -> Problema NP completo (características de la tarea y dinámica naturaleza de los recursos heterogéneos)

Task scheduler recibe tareas -> las mapea a los recursos disponibles, tomando en consideración características de la tarea y los parámetros del recurso.

Por ello, se busca un algoritmo óptimo de task scheduling para lograr una **eficiente y buena utilización de recursos con la máxima ganancia y rendimiento**.

Métodos meta-heurísticos

Basados en población:

PSO (optimización por enjambre de partículas)

AS (Sistema de Hormigas) -> Elistas, Rankings

Basados en trayectoria

SA (buscar un vecino más cercano aleatoriamente y si es mejor entonces se considera).

Trabajos relacionados

- Muchos no consideran **características de la tarea y propiedades de recursos dinámicamente**.
- Se centran en la optimización o el tiempo de respuesta sin considerar **balanceo de carga ni utilización de recursos**.
- Otros no consideran el **tiempo de espera de tareas**.
- No consideran **características de recursos** con tareas conjuntamente en el cálculo de la función fitness.
- Otros solo funcionan para **casos específicos no generales**.

¿Qué se utilizará para resolver el problema en este paper?

Primero para optimizar el tiempo de espera y disminuir el tamaño de la cola:

WTO(waiting time optimizacion) (PSO y AS)

Luego,se utilizaran colas de despacho dinámico (TSDQ) combinadas con:

FLPSO = Fuzzy Logic + PSO

SAPSO = SA + PSO

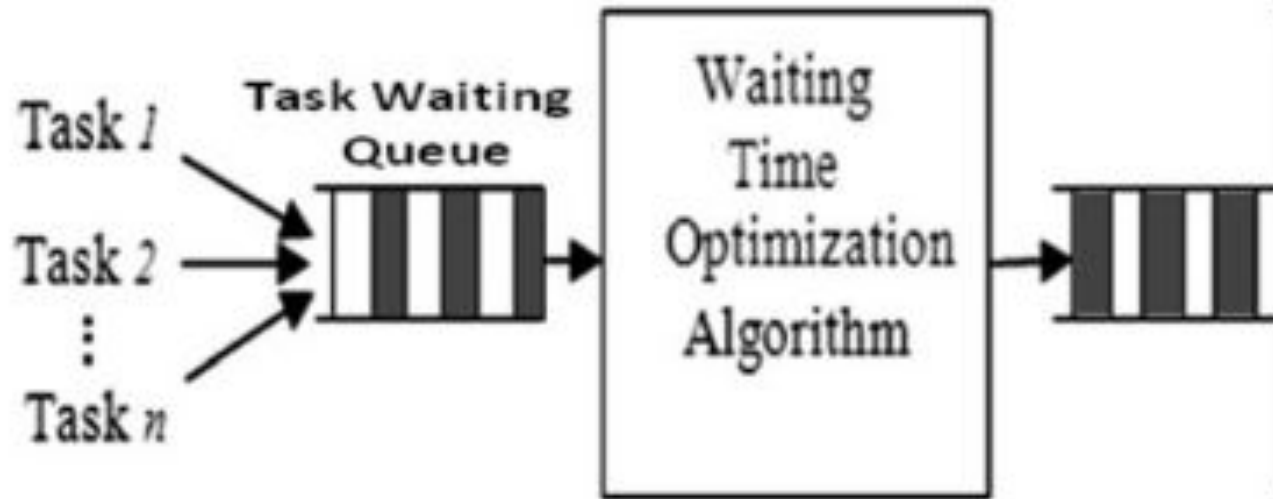
Propuestas

- Incrementan la convergencia de PSO (Simulated annealing y Fuzzy Logic).
- Incorporan características de las tareas y capacidad de los recursos.
(RAM, MIPS, CPUs).
- Considera características dinámicas del espacio buscado.
(Diferentes características de VMs)

Objetivos

- **Minimizar makespan**
- **Minimizar costo de ejecución**
- **Maximizar utilización de recursos**
- **Buen balanceo de carga**

WTO-PSO



Simulated Annealing

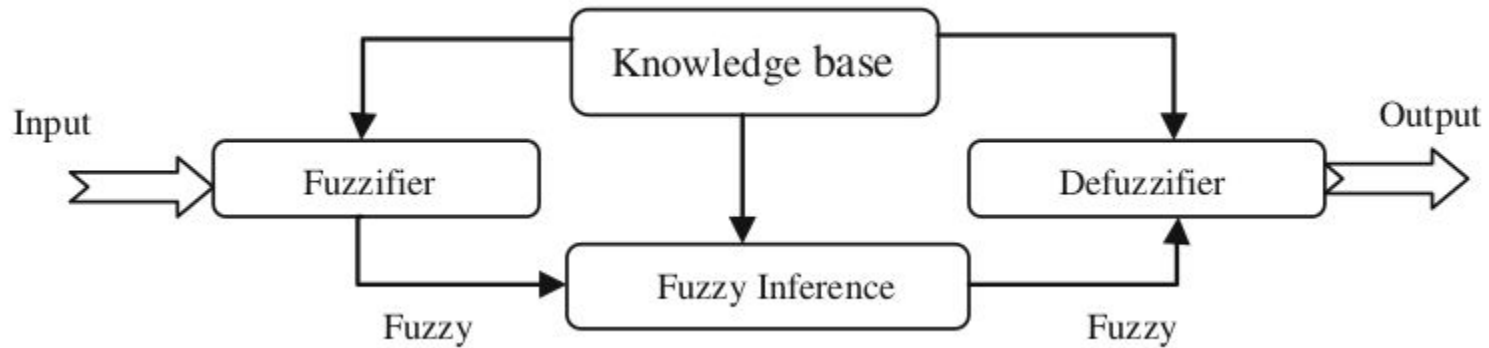
Algorithm 3. Pseudo-code of Simulated Annealing (SA)

```
1. S0 = Generate_initial_solution()
2. Sbest = S0
3. Temperature = Initialize_Temp()
4. repeat
5.     repeat
6.         Temperature = calculate_temperature()
7.         Snew = Create_neighbor_solution(S0)
8.         if (Fitness(Snew) < Fitness(S0)) then
9.             Sbest = Snew
10.        else if (rand() <  $e^{\frac{Fitness(S_0) - Fitness(S_{new})}{Temp}}$ )
11.            S0 = Snew
12.        end if
13.    until thermal equilibrium is reached
14.    decrease Temperature according with the annealing schedule;
15. until stopping criterion is met
16. return Sbest
```

Lógica Difusa

Task length	CPU speed	RAM	Status of VM	Result
Low	Low	Low	Low	High
Low	Low	Medium	Low	High
Low	Medium	Low	Medium	Medium
Low	Medium	Medium	High	Medium
Low	Medium	High	Medium	Medium
Low	High	Medium	Low	High
Low	High	Medium	Medium	Medium
Low	High	High	High	Low
Medium	Low	Low	Low	High
Medium	Low	Medium	High	Low
Medium	Low	High	Medium	Medium

Lógica Difusa



Propuesta - Consideraciones

1. **La independencia de las tareas.**
2. **No hay preferencias.**
3. **No pueden ser interrumpidas.**
4. **Características requeridas (length, arrival time, burst time, deadline,etc)**
5. **Heterogeneidad de las VMs (diferentes en CPU speed, RAM, bandwidth,etc)**

Propuesta - Problemática

1. Las tareas entrantes de unen a la primera cola de espera mientras se utilizan recursos en otras.
2. No solo satisfacer QoS, sino balanceo de carga de VMs (mejorar utilización de recursos y maximizar ganancia del proveedor).

Arquitectura de la propuesta

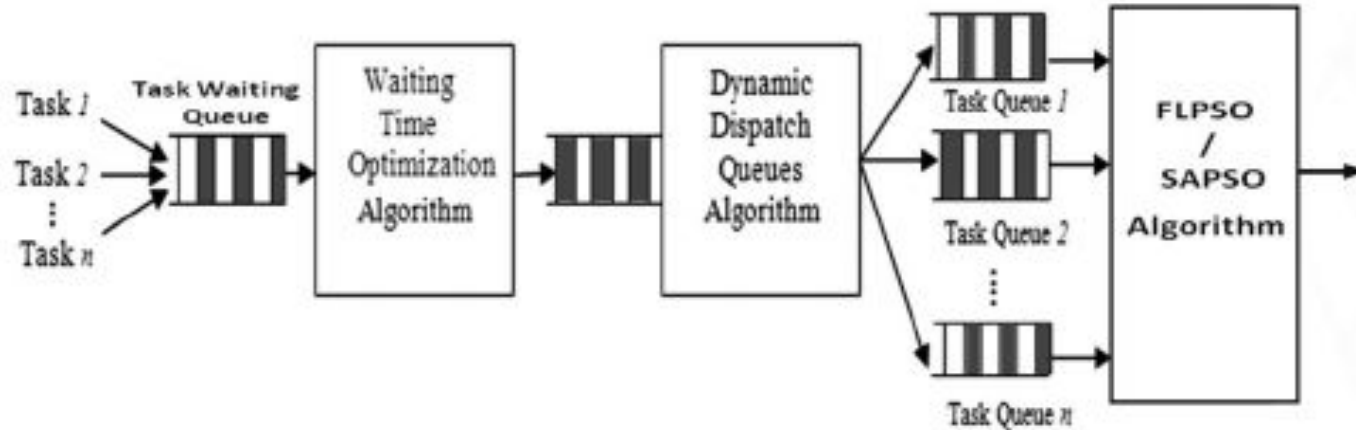


Fig. 3 The proposed approach model

TSDQ - Dynamic Decision Queues Algorithm

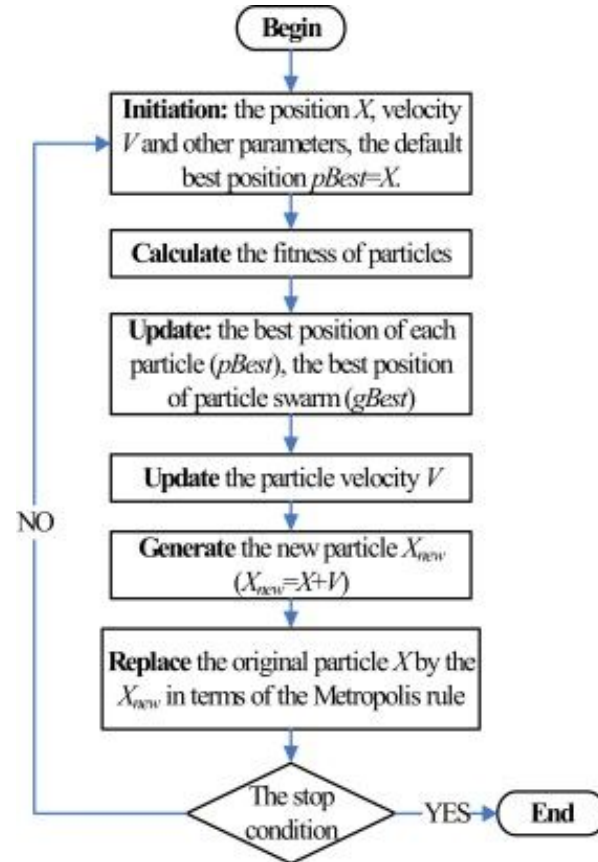
Realiza decisiones con los mejores caminos posibles para crear un óptimo número de colas, lo que garantiza el schedule de tareas eficientemente.

NOTA: Se considera el máximo de colas generadas como el número de número de máquinas virtuales.

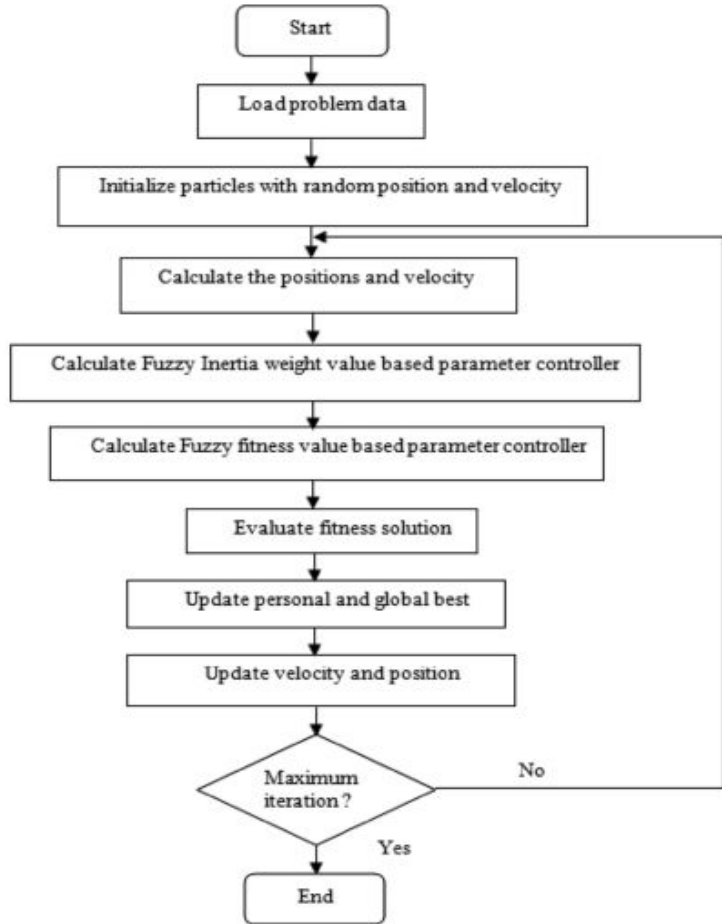
FLPSO y SAPSO

- 1) Basados en la población para buscar en toda la nube de posibilidades
- 2) Debe ser rápido para adaptarse a las características dinámicas de la nube.
- 3) Se utiliza FL y SA para hallar una respuesta más rápida que el PSO normal.

Algoritmo SAPSO



Algoritmo FLPSO



Experimentación 1 - Tiempo de Espera

FCFS versus WTO-PSO

Table 3 Tasks characteristics

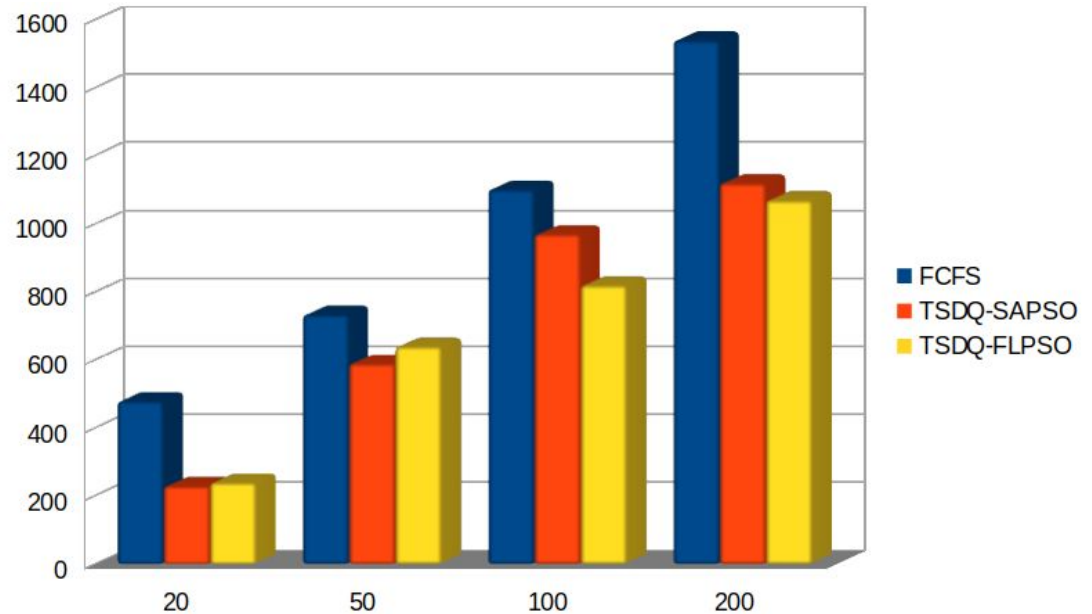
Task ID	Task burst time	Task length
1	10	800
2	30	100
3	13	3500
4	19	3000
5	44	2326
6	4	3300
7	8	4000
8	36	5000
9	20	6000
10	50	7000

Experimentación 1 - Resultados

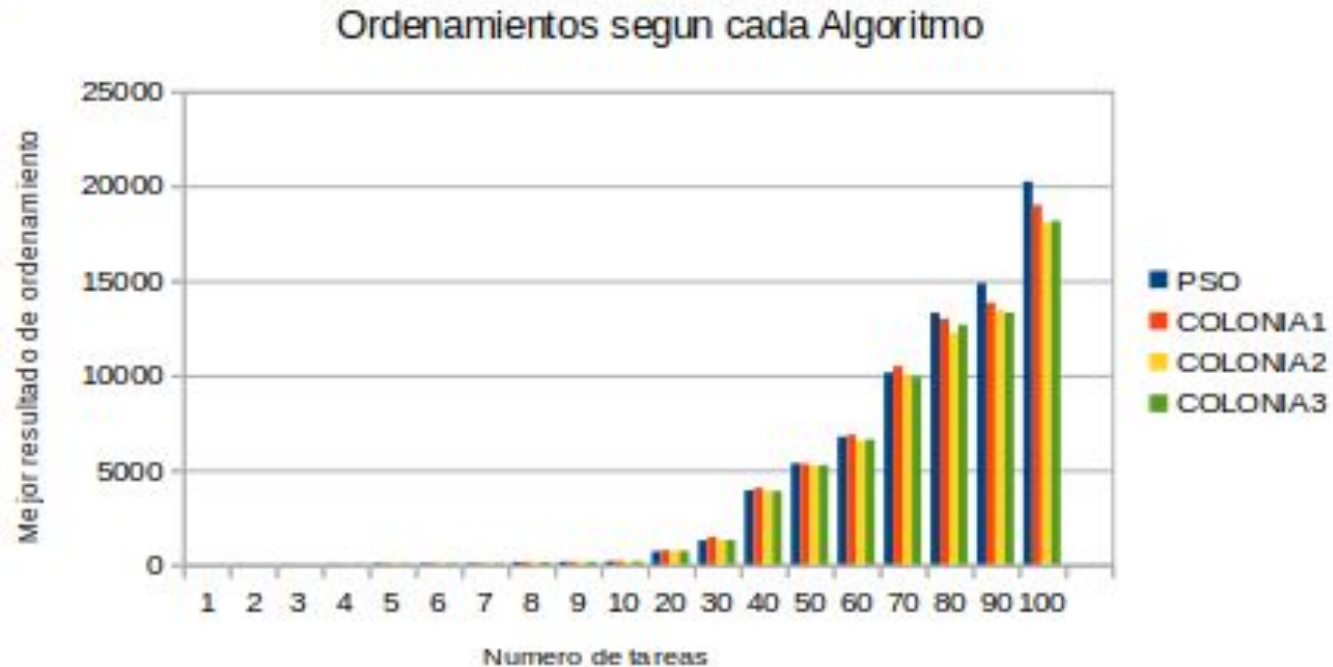
	Task1	Task2	Task3	FCFS	WTO-SO
1	6	4	8	5,33	4,6666
2	34	12	20	26,6666	14,6666
3	43	15	30	33,6666	20
4	24	25	10	24,3333	14,6666
5	13	5	20	10,3333	7,6666
6	30	12	18	24	14
7	40	20	33	33,3333	24,3333
8	14	24	21	17,3333	16,3333
9	8	3	14	6,3333	4,6667
10	16	10	14	14	11,3333
				195,33	132,333

Experimentación 2 - Comparación Makespan

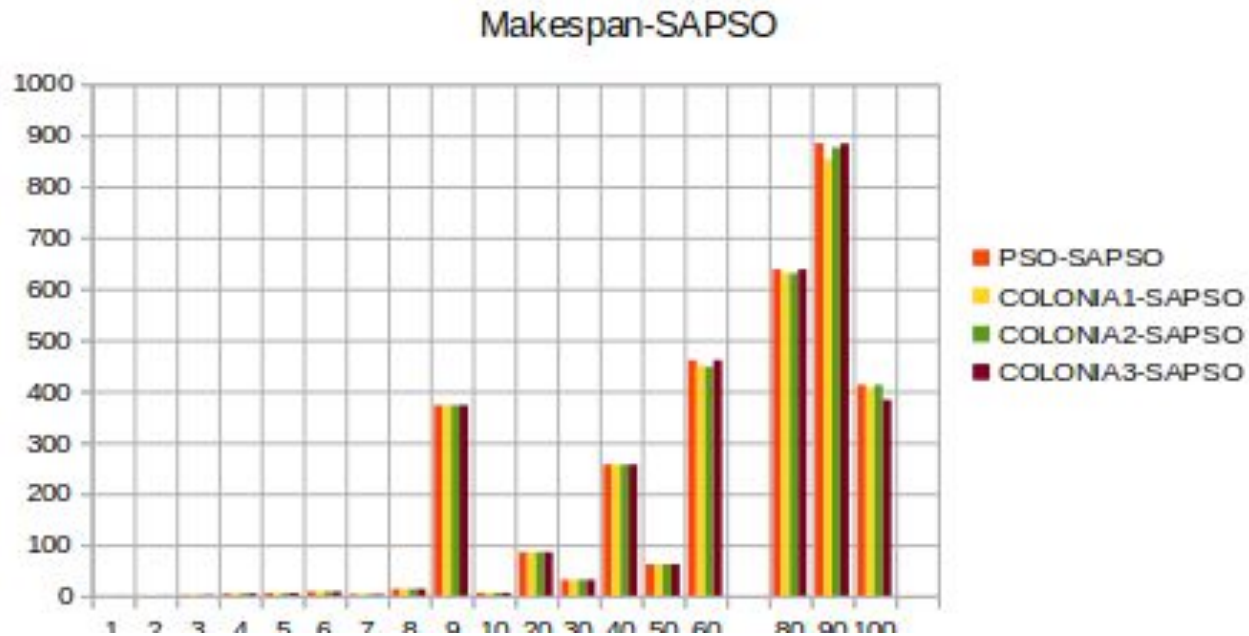
Makespan : es un término usado en “scheduling” y representa la forma de organizar tareas a trabajadores de tal manera que acabe en el menor tiempo posible.



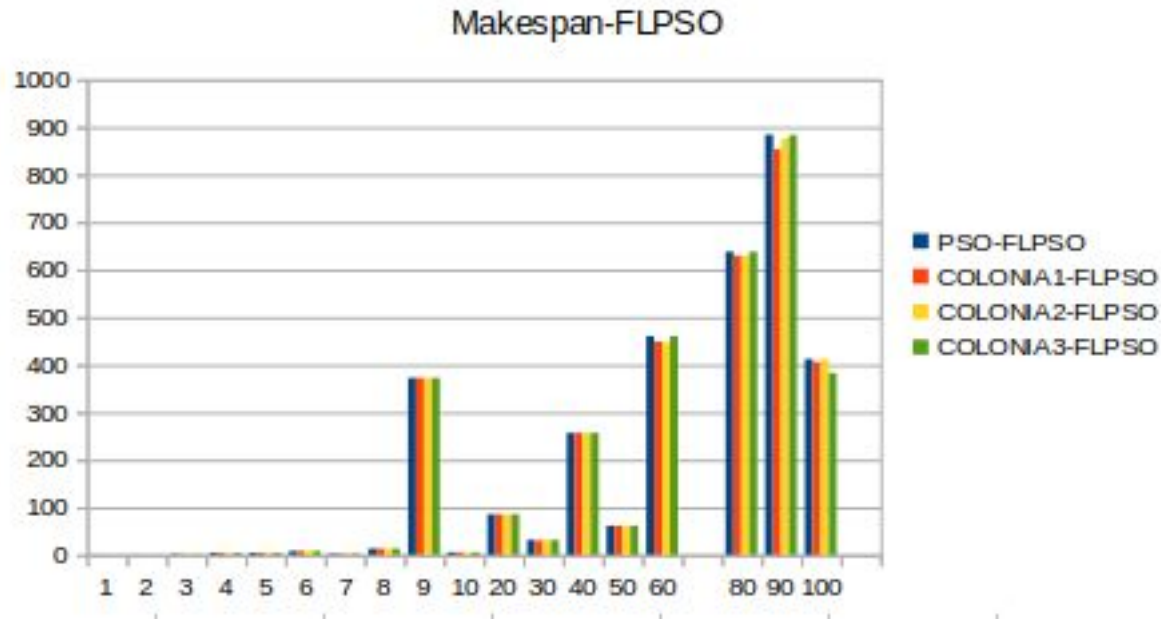
Experimentación 2.1 - Ordenamientos



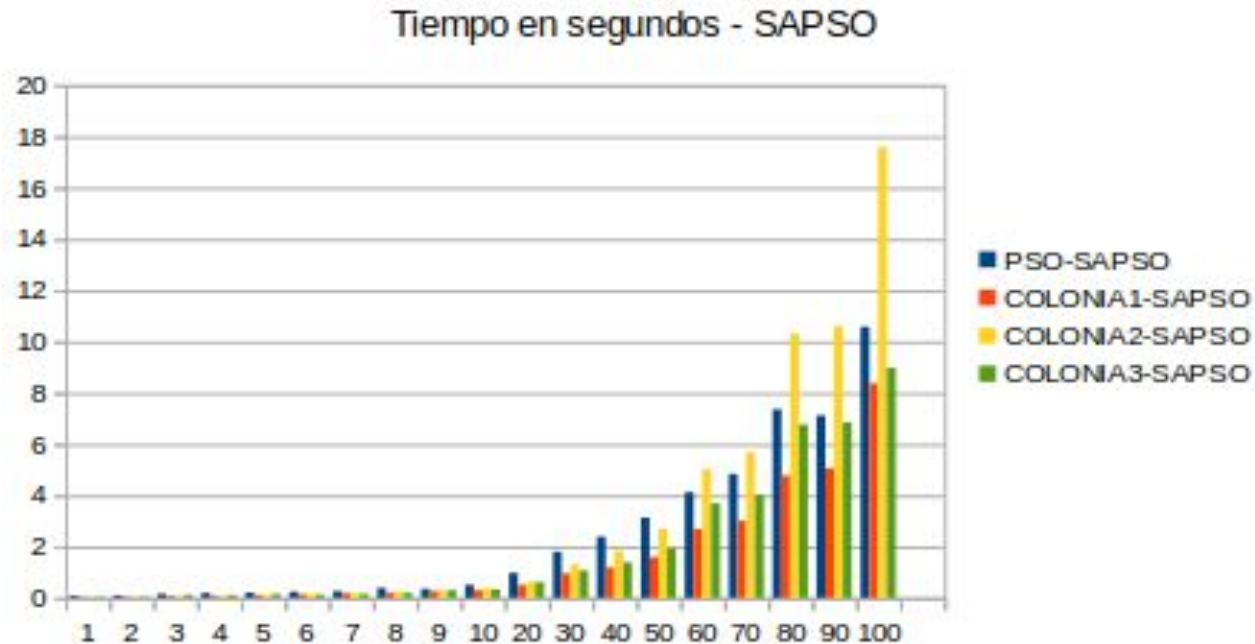
Experimentación 2.2 - Makespan SAPSO



Experimentación 2.3 - Makespan FLPSO



Experimentación 2.4 - Eficiencia SAPSO



Experimentación 2.5 - Makespan FLPSO

