An Integrated Algorithm Method to Optimize Resource Allocation with a Case Study of Production Line

Group 4

105034534 Becky Dong 105034541 Neal Lee

105034542 Mavies Liao 105034556 Kevin Jao

Agenda

1. Introduction
2. Literature Review
3. Methodology
4. Case Study
5. Analysis & Discussion
6. Conclusion & Recommendations

Introduction

1. *Case*
2. *Aim*
3. *Method*
4. *Step*

Introduction

- Case: A continuous flow of production line
- Aim: Optimize the allocation of machines and buffers in a production line for increasing profits.
- Method: Simplified Swarm Optimization (SSO) and Simulated Annealing (SA) .

Introduction

- Step 1: Combination of Simplified Swarm Optimization (SSO) with Simulated Annealing (SA) .
- Step 2: Compare the effect of the results.
- Step 3: Verify the results.

Literature Review

Simplified Swarm Optimization

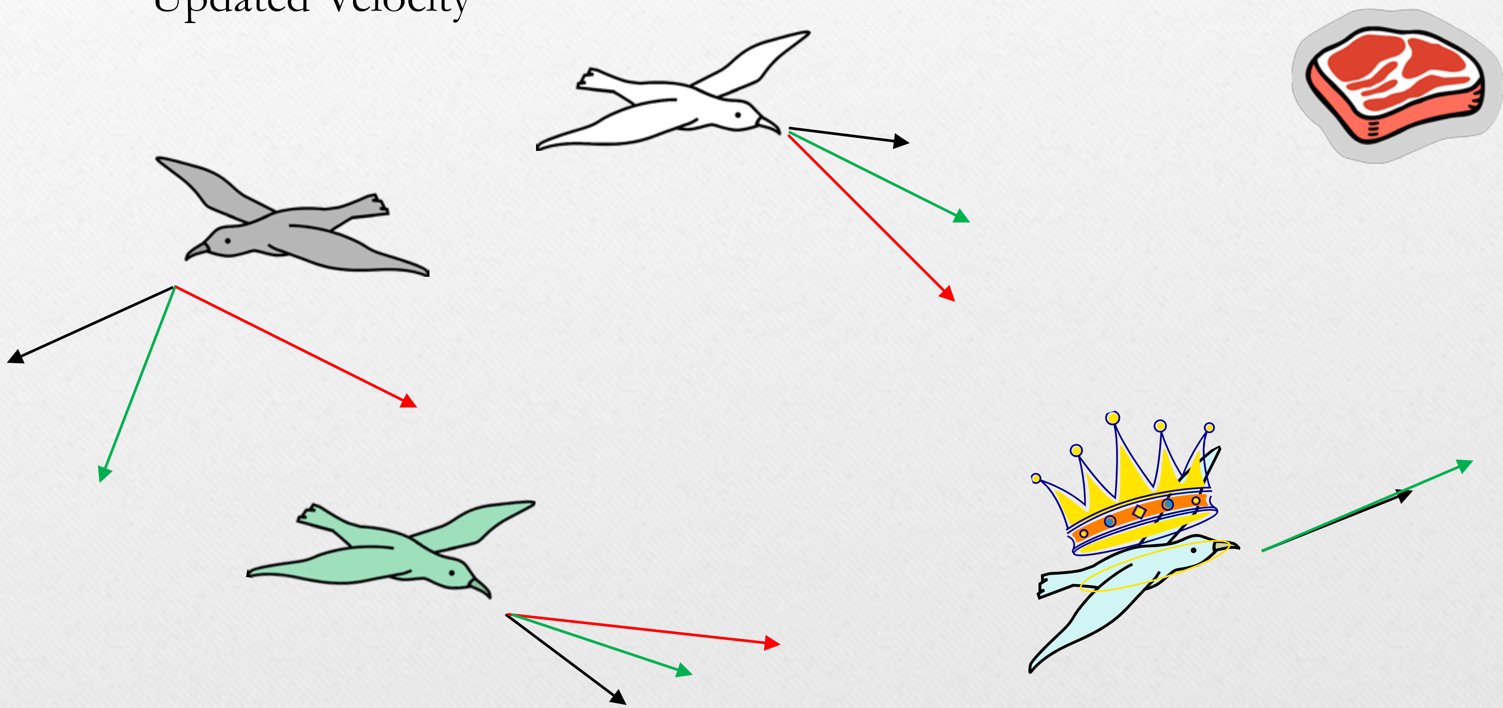
- Simplified Swarm Optimization (SSO) was originally proposed by Yeh in 2009.
- SSO is also called the discrete PSO(Particle Swarm Optimization).
- SSO is a stochastic optimization algorithm to compensate for the drawbacks of PSO in solving discrete problems.
- However, this algorithm is easily reflected by initial solution.

PSO Searching Example (1/2)

Original Velocity

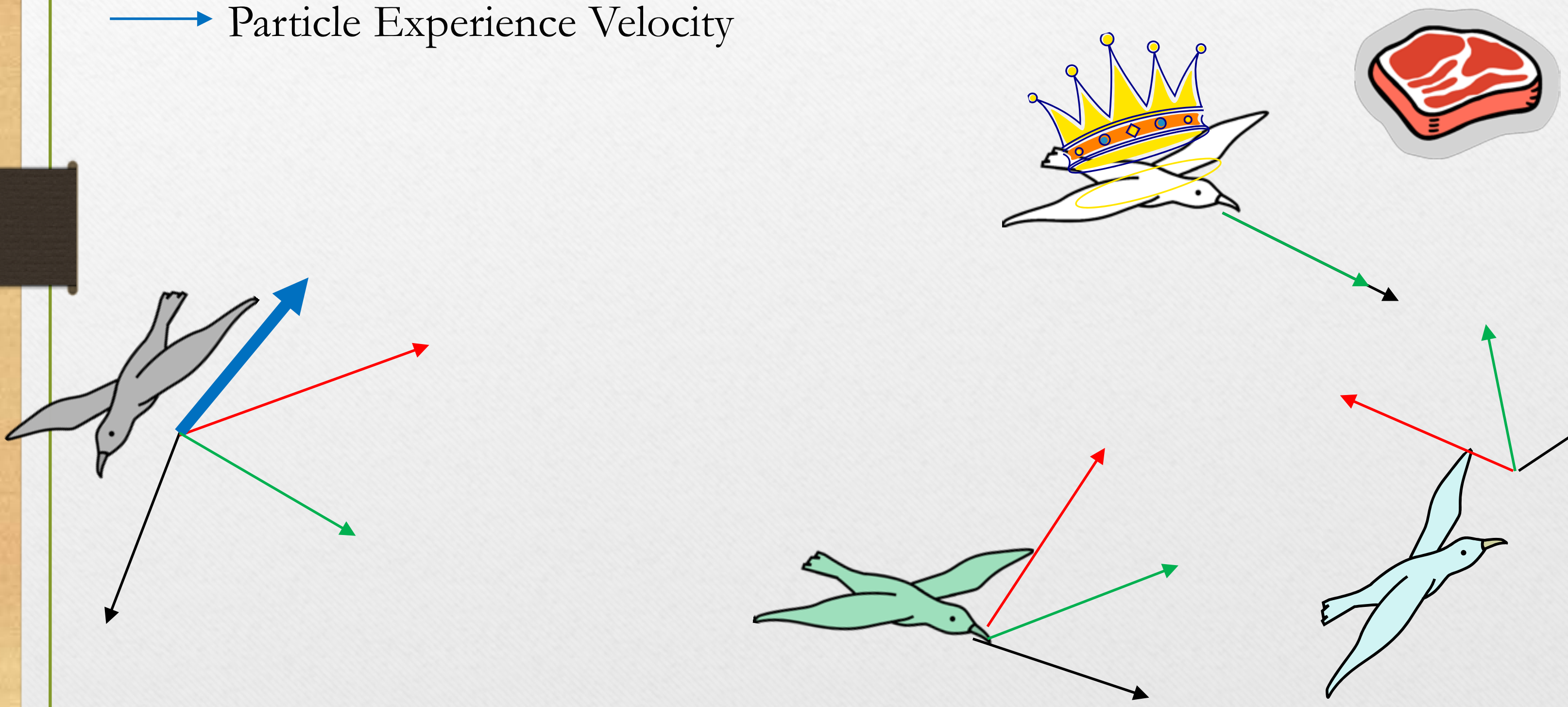
Global Experience Velocity

Updated Velocity



PSO Searching Example (2/2)

- Original Velocity
- Global Experience Velocity
- Updated Velocity
- Particle Experience Velocity



PSO Equation

1. **Update Velocity Equation**

$$\overrightarrow{V_{i,new}} = w \cdot \overrightarrow{V_{i,old}} + c_1 \cdot rand() \cdot (P_i - X_{i,old}) + c_2 \cdot rand() \cdot (G - X_{i,old})$$

2. **Update Position Equation**

$$X_{i,new} = X_{i,old} + \overrightarrow{V_{i,new}}$$

Notation :

W ： Weight

C1 ： Individuality variable

C2 ： Sociality variable

P ： Particle best

G ： Global best

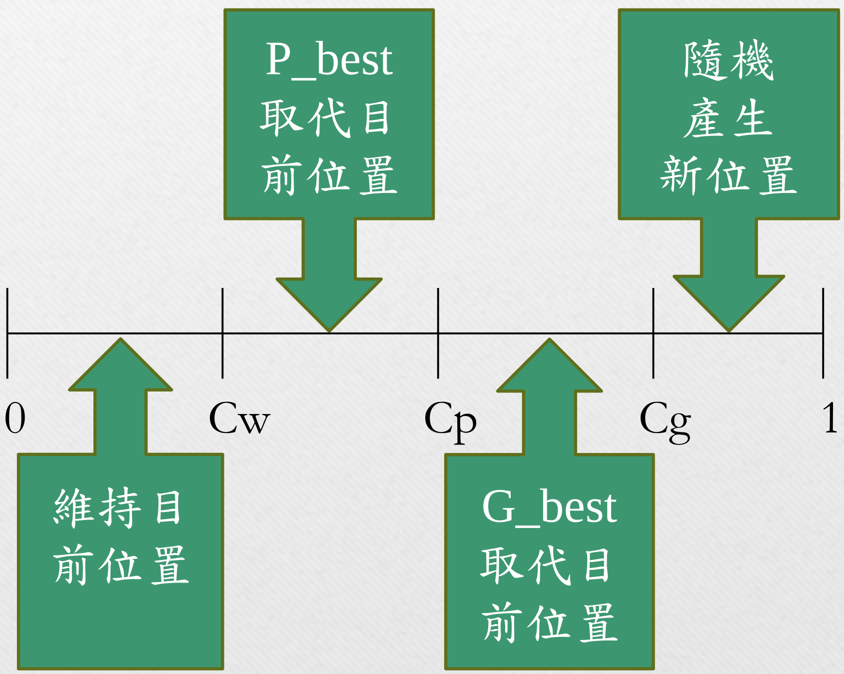
SSO Equation

$$x_{id}^{t}=\begin{cases} x_{id}^{t-1}, & \text{if } rand() \in [0, C_w), \\ p_{id}^{t-1}, & \text{if } rand() \in [C_w, C_p), \\ g_{id}^{t-1}, & \text{if } rand() \in [C_p, C_g), \\ x, & \text{if } rand() \in [C_g, 1). \end{cases}$$

$i = 1, 2, \ldots, m$, m : the swarm population.

$X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$,

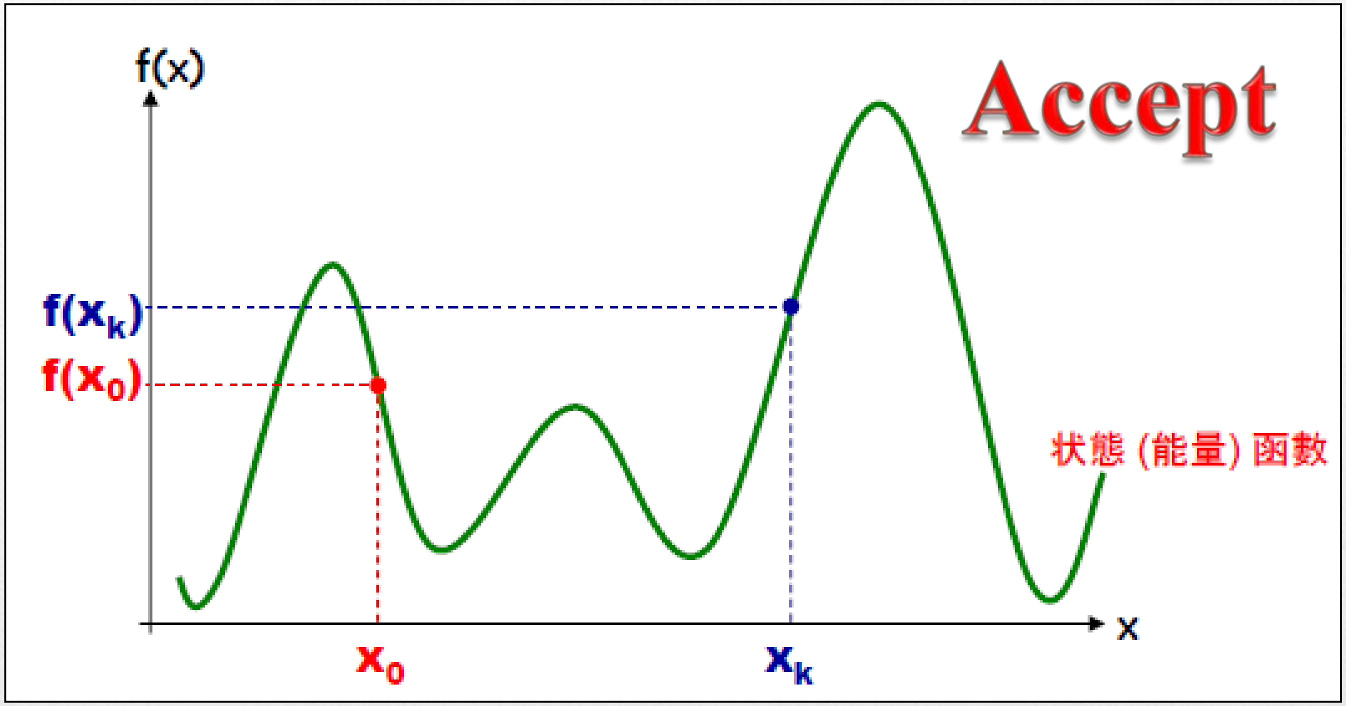
x_{id}^{t}: the D-th dimension of i-th particle in generation t



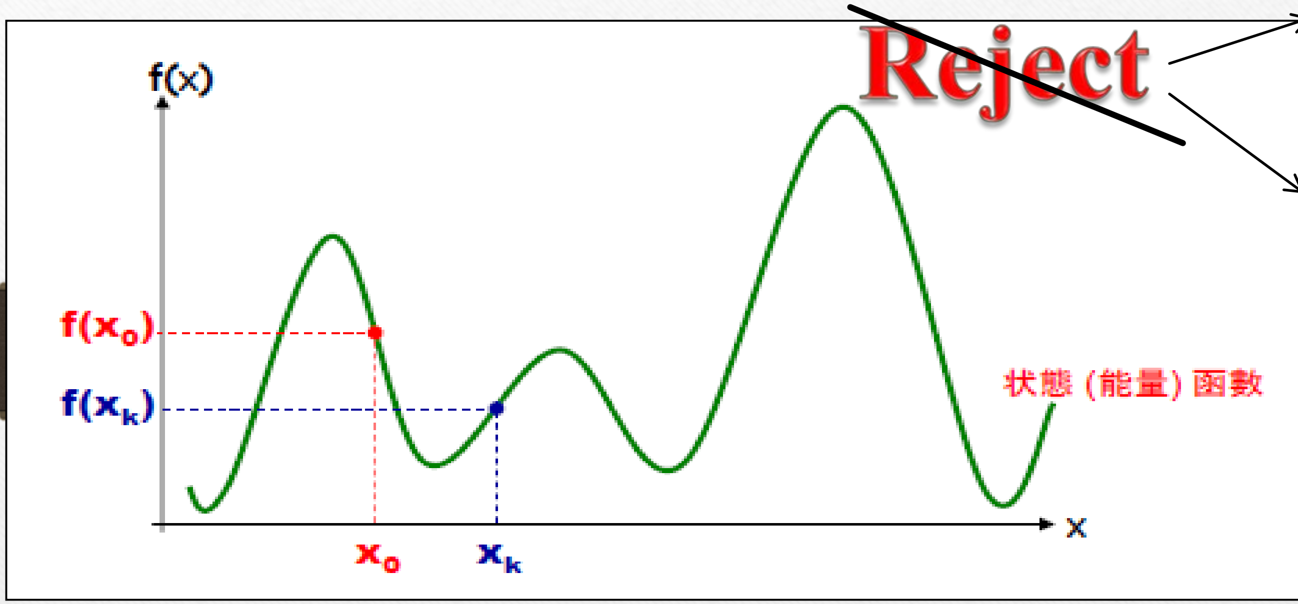
Simulated Annealing

- Simulated Annealing (SA) was originally proposed by S.Kirkpatrick in 1980.
- SA describes a group of heuristic optimization techniques based on iterative improvement
- SA is motivated by an analogy to the statistical mechanics of annealing in solids.
- SA cannot identify whether it has found an optimal solution

Simulated Annealing Searching Method(1)



Simulated Annealing Searching Method(2)



If $R \le PR(A)$ ➡ Accept

If $R \le PR(A)$ ➡ Reject

$$PR(A) = \min\{1, e^{\left(\frac{-\Delta f}{T_k}\right)}\}$$

Notations

PR(A)是接受狀態xk之機率

Δf = f(xk) - f(xk-1)

Tk是指第k個狀態的Control Temperature

R : random number

Comparison of SSO and SA

	Advantage	Disadvantage
SSO	(1) It is suitable for solving discrete problems. (2) Update mechanism is simpler than PSO.	(1) It is easily influenced by initial solution.
SA	(1) It is relatively easy to code, even for complex problems. (2) The implement time is short.	(1) The method cannot tell whether it has found an optimal solution. (2) It only has one particle

Methodology

Methodology

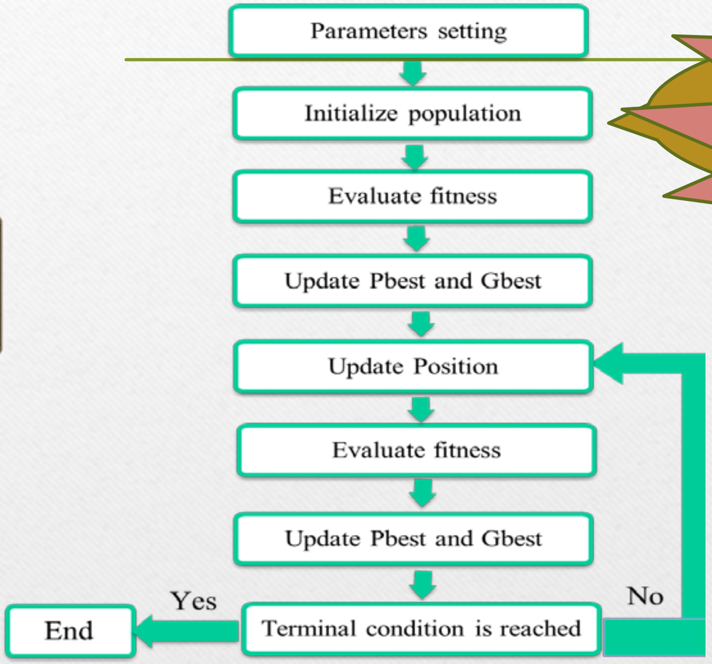


Fig1. The flowchart of SSO.

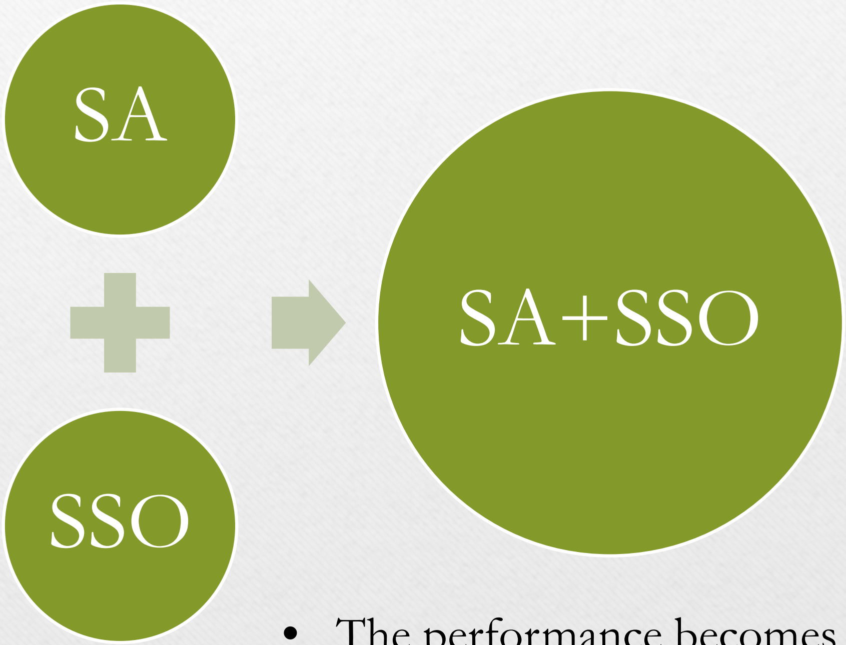
Use SA

SSO is easily influenced by initial solution.

Methodology

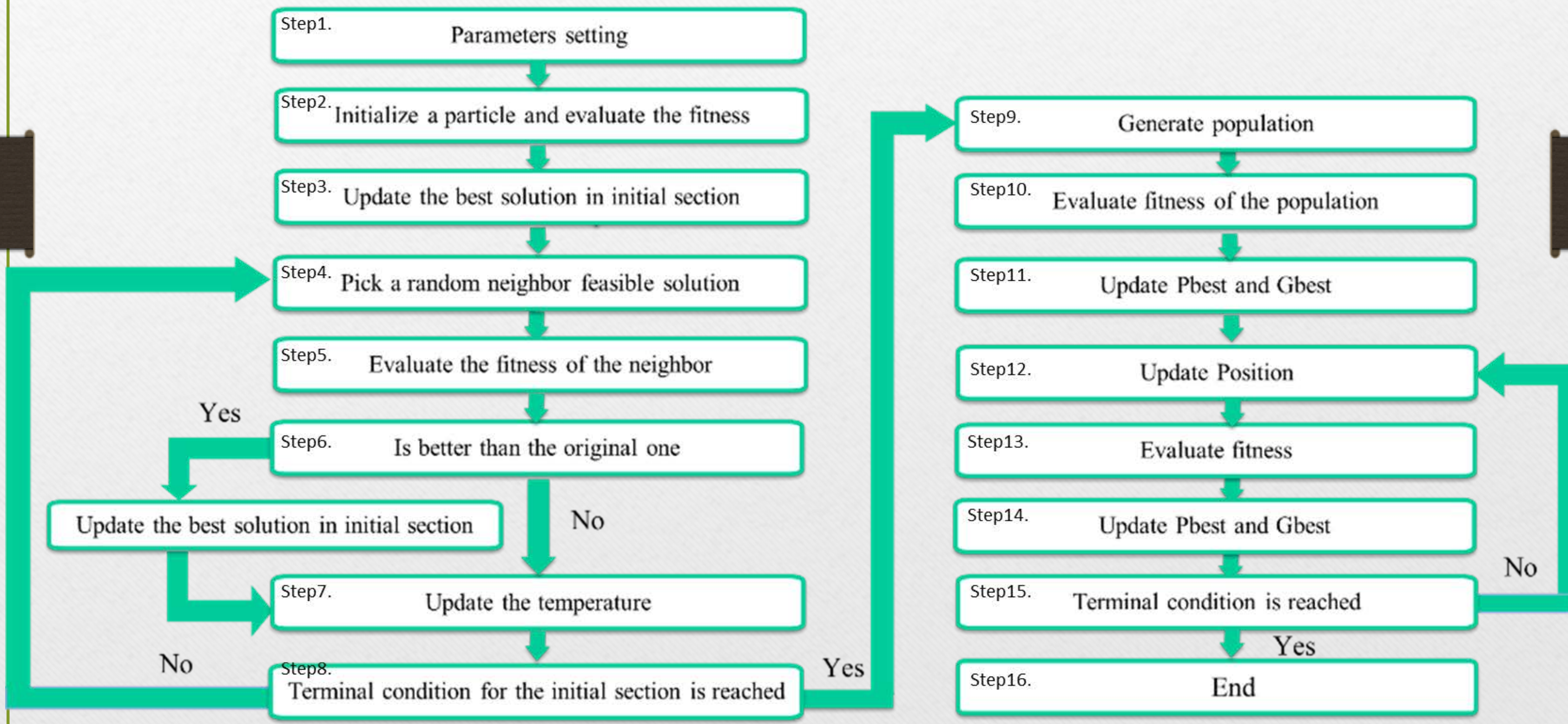
- Find better initial solution.
- Implement time is short.

- It is suitable for solving discrete problems.



- The performance becomes better, and find the best solution more efficiently

Methodology



The Mathematical Model

- Notations

  W_i number of machine in workstation i, $1 \leq i \leq m$.

  B_j size of capacity in buffer j, $1 \leq j \leq n$.

  TH throughput of the production line

  M_c cost of operating for each machine.

  B_c cost of buffer for each capacity.

  P price of each product

- Objective function:

  $Max\ Profit = P*E[TH] - M_c \sum_{i=1}^{m} W_i - B_c \sum_{j=1}^{n} B_j$

- Subject to：

  $1 \leq W_i \leq 3, \quad \forall i$

  $1 \leq B_j \leq 10, \forall j$

Case Study

1. *A production line*
2. *Aim*
3. *assumption*

Case Study

- A production line
  - four parts
  - three finite-size buffers
  - an infinite supply of blank parts.

- Aim: Find the optimal number of machines and buffers for maximizing the profits.

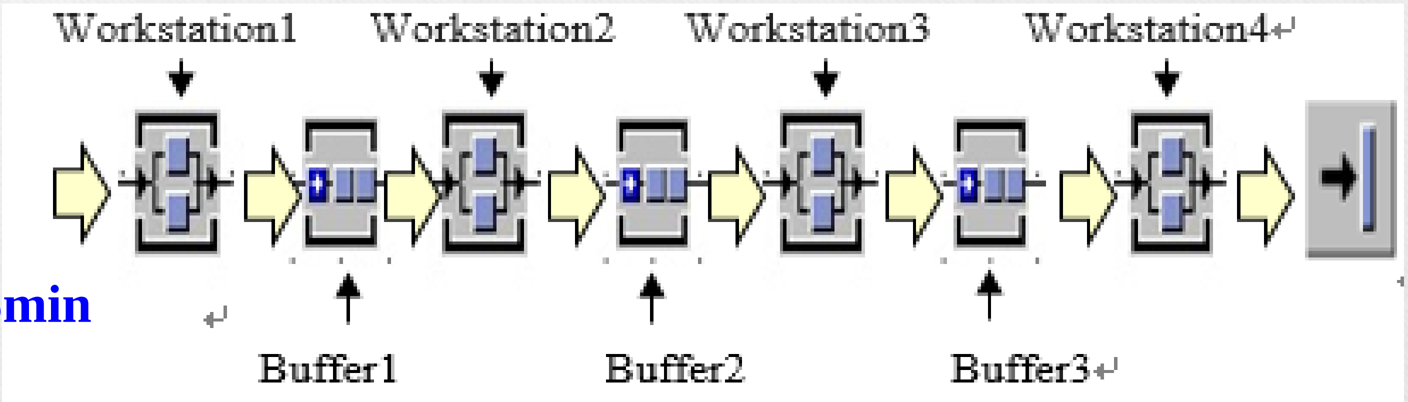
Assumption

- Costs
  - Buy one machine:$25,000
  - Add one buffer: $5,000
- Revenue
  - Sell one product: $100

Assumption

Materials arrival and process time follow exponential distribution.

λ= 20min λ= 30min λ= 12min λ= 15min



λ= 5min

Fig 2. Production Line

Assumption

A. Objective Function

$$Max\ Revenue = P*E[TH]-M_c \sum_{i=1}^{m} W_i - B_c \sum_{j=1}^{n} B_j$$

B. Constraints

$$1 \leq W_i \leq 3, \quad 1 \leq i \leq 4,\ i \in N.$$

$$1 \leq B_j \leq 10, \quad 1 \leq j \leq 3,\ j \in N.$$

C. Parameter

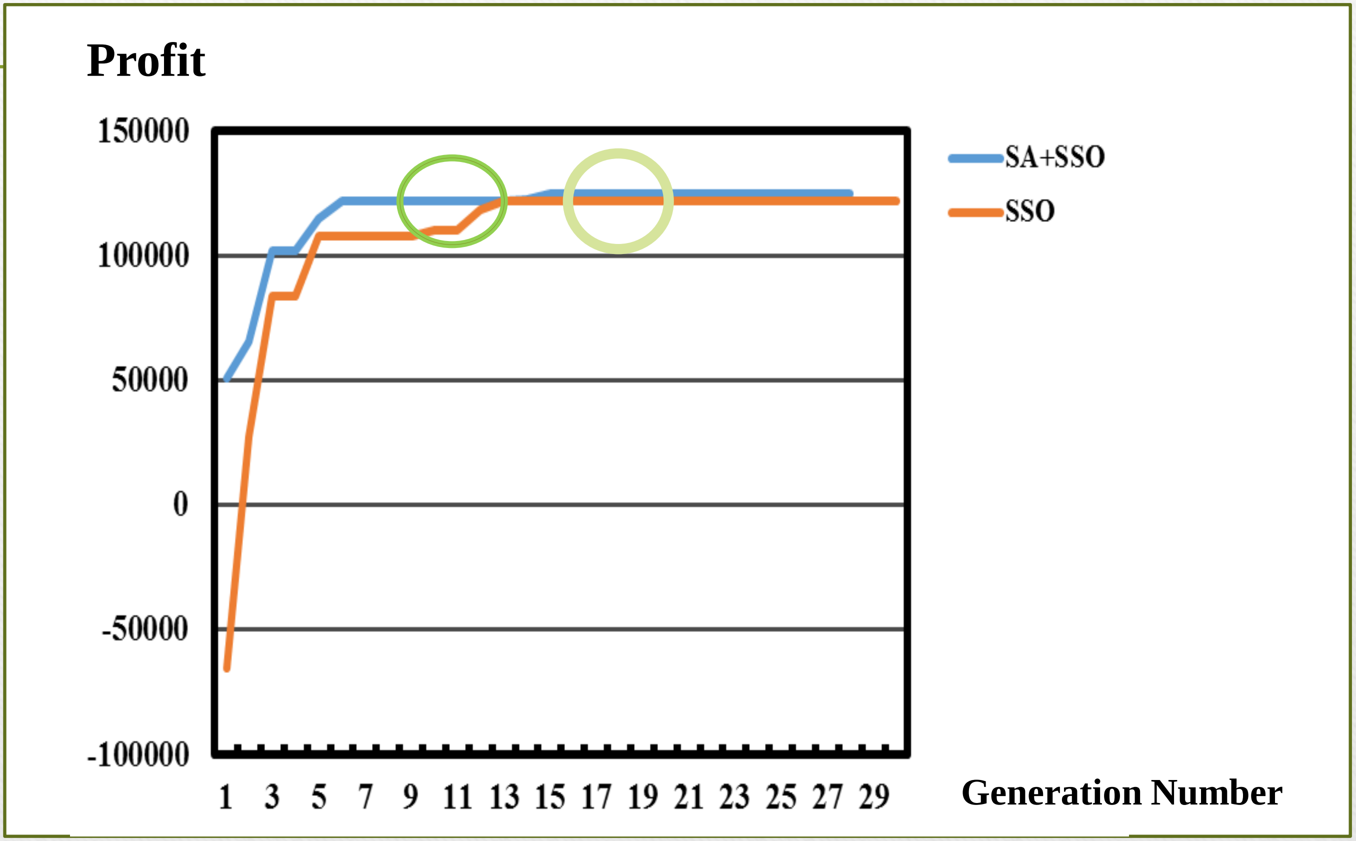
M_c=25000

B_c=5000

P =100

Analysis & Discussion

Optimal solution for each iteration



Simulation time of two methods

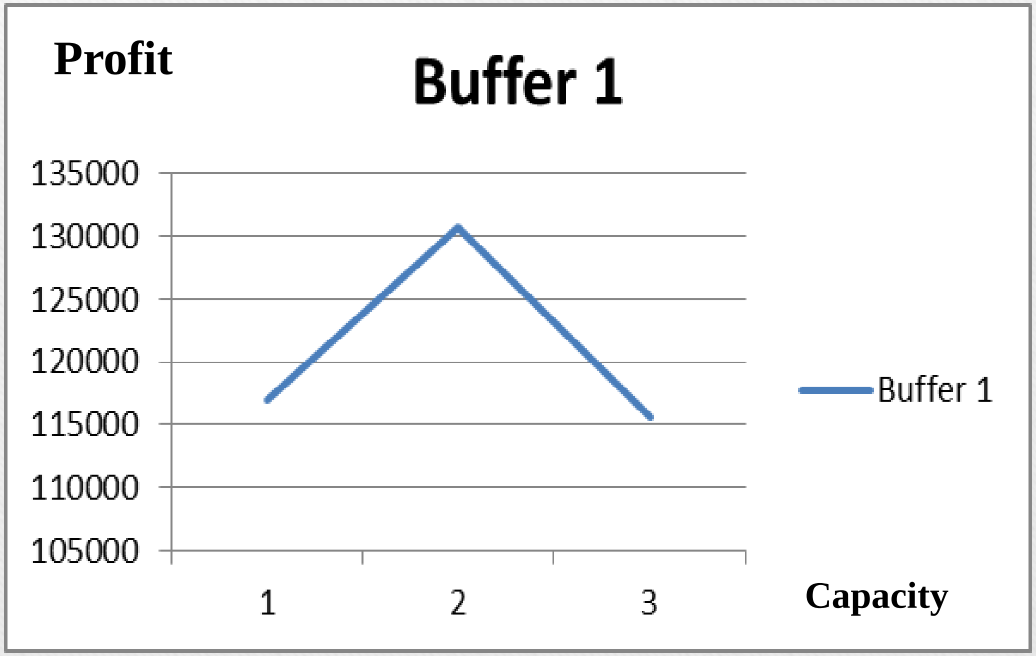
Method	SA+SSO	SSO
Simulation time	8.142	10.83

Optimal solution for resource allocation

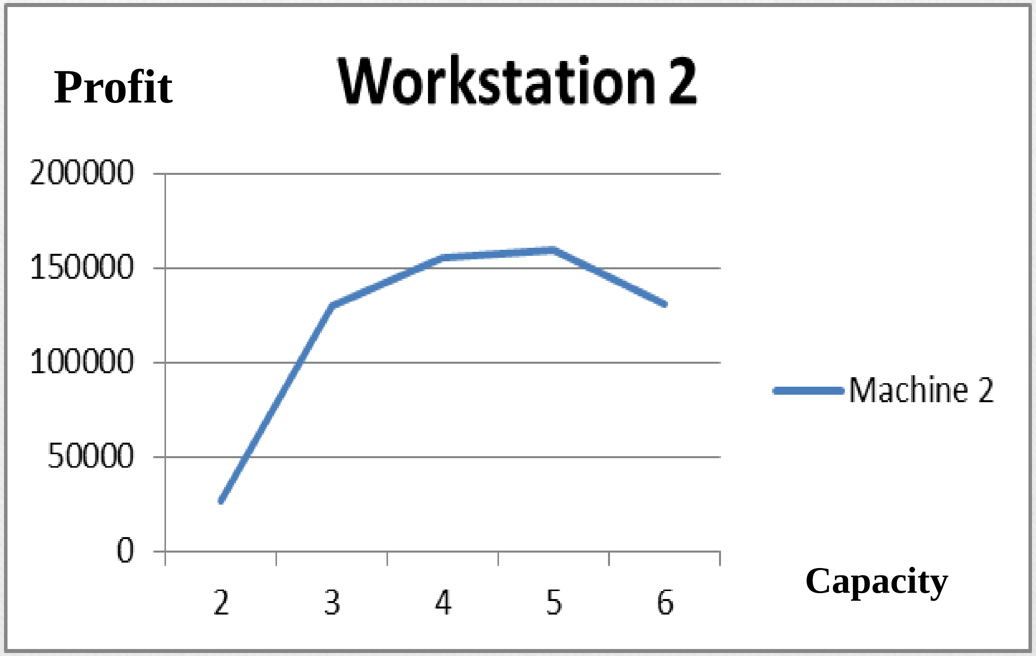
Variable	W_1	W_2	W_3	W_4	B_1	B_2	B_3
Number	3	3	2	2	2	3	1

Sensitivity analysis

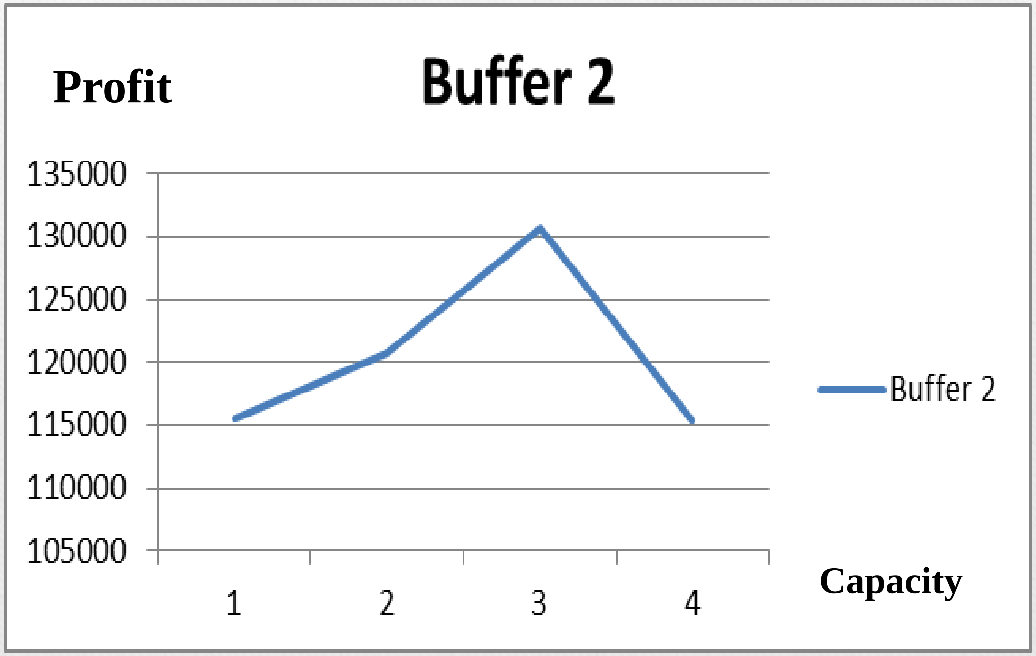
Sensitivity analysis(1)



Sensitivity analysis(2)



Sensitivity analysis(3)



Conclusion & Recommendations

Conclusion & Recommendations

Conclusion

- This integrated method overcomes the drawbacks of SSO, which is easily influenced by the initial solution.
- It could find the optimal solution more efficiently using lower simulation resources and time.

Recommendations

- the integrated method is expected to be used in the problems with larger feasible solution region.

Q & A

Thanks for listening.