




# Aplicación del Algoritmo de Búsqueda Gravitacional para Optimizar un Problema de Planeación Agregada de la Producción

## Gravitational Search Algorithm Application to Optimize an Aggregate Production Planning Problem

Alberto Ortiz-Licona<sup>a</sup> , Marco Antonio Montúfar-Benítez<sup>a</sup> , Juan Carlos Seck Tuoh-Mora<sup>a</sup> 

<sup>a</sup> Área Académica de Ingeniería, Universidad Autónoma del Estado de Hidalgo, 42184, Pachuca, Hidalgo, México.

### Resumen

Dada la necesidad de buscar alternativas que permitan encontrar diversas soluciones al problema de planeación agregada de la producción (APP), en esta investigación se propone una solución por medio del algoritmo de búsqueda gravitacional (GSA). Se incluye el manejo de restricciones y, así mismo, se realiza una introducción a las funciones de desempeño a las que ha sido sujeto, que permiten mostrar que el algoritmo tiene capacidad de encontrar soluciones óptimas a problemas con alto grado de dificultad; finalmente se exponen los resultados de la propuesta demostrando que este algoritmo de búsqueda y optimización puede ser aplicado para solucionar este tipo de problemas donde se tiene una gran cantidad de variables y restricciones.

### Palabras Clave:

Algoritmo de Búsqueda Gravitacional, Optimización, Computación Evolutiva, Planeación Agregada de la Producción, MATLAB.

### Abstract

Given the need to find alternatives that allow finding various solutions to the problem of aggregate production planning (APP), this research proposes a solution through the gravitational search algorithm (GSA). Constraint management is included and, likewise, an introduction is made to the benchmark functions to which it has been subjected, which allow to show that the algorithm has the ability to find optimal solutions to problems with a high degree of difficulty; Finally, the results of the proposal are shown, demonstrating that this search and optimization algorithm can be applied to solve these types of problems where there are a large number of variables and constrictions.

### Keywords:

Gravitational Search Algorithm, Optimization, Evolutionary Computing, Aggregate Production Planning, MATLAB.

## 1. Introducción

La planeación de actividades en una compañía se puede clasificar en tres niveles jerárquicos que son: estratégico, táctico y operativo. Tales niveles se enfocan en largo, mediano y corto plazo, respectivamente. Los niveles táctico y operativo tienen como objetivo planear y programar el uso de los recursos, con lo que generan valor agregado a las actividades productivas de la organización. El horizonte típico de planeación se encuentra entre 3 y 15 meses, pero la mayoría de compañías establece el horizonte en 12 meses con el fin de desarrollar un programa anual de planeación agregada de la producción (APP) y así prever situaciones inesperadas. En

otras palabras, el plan deberá ser usado para prevención de fluctuaciones en la demanda, las cuales están directamente relacionadas al exceso o falta de recursos disponibles en la organización.

La APP permite conocer simultáneamente los niveles óptimos de producción, inventario y empleo en un horizonte de planificación finito dado para satisfacer la demanda total de productos que comparten los mismos recursos limitados (Buffa, 1972). Así mismo, brinda a la administración una idea sobre la cantidad de materiales y otros recursos que se deben adquirir y cuándo, de modo que el costo total de las operaciones se mantenga al mínimo durante ese periodo. Los costos típicos incluyen costos por trabajadores contratados y

\*Autor para la correspondencia: alortiz.mx@gmail.com

Correo electrónico: alortiz.mx@gmail.com (Alberto Ortiz Licona), montufar@uaeh.edu.mx (Marco Antonio Montufar Benítez), jseck@uaeh.edu.mx (Juan Carlos Seck Tuoh Mora)

despedidos, costos de producción, artículos subcontratados, horas extra e inventarios de cada periodo (Al, 2012), (Da Silva, 2006).

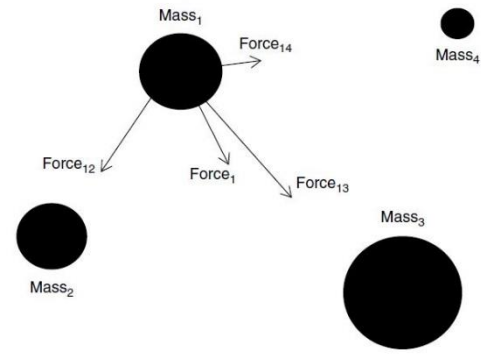
Las tendencias indicadas en el trabajo de (Cheraghalikhani, 2019) brindan a esta investigación un amplio campo de relevancia debido a que la solución a este tipo de problemas aplicando metaheurísticas continúa siendo de gran relevancia. La presente investigación propone la solución de un problema planeación agregada de la producción por medio de un algoritmo de búsqueda gravitacional (GSA). Se exponen los resultados de la aplicación de esta técnica y son comparados con la solución encontrada por LINGO.

### 1.1. Literatura sobre APP

Se han introducido modelos de APP con diferentes grados de sofisticación desde la década de los 50's con los trabajos de (Holt, 1955), (Holt, 1956), donde destacan la importancia de este problema. Calcularon una solución óptima generalizada, basada en una técnica conocida como regla de decisión lineal (LDR). Su modelo fue aplicado en una fábrica de pintura para generar un plan de producción. En el trabajo de (Nam, 1992) se estudiaron los principales modelos y métodos conocidos en los años 80's y 90's, donde se encuentran técnicas basadas en métodos gráficos hasta las más sofisticadas búsquedas, en general clasifican las técnicas en dos grandes grupos: los que garantizan encontrar una solución óptima exacta y los que no lo hacen.

En las últimas décadas, el problema de APP se ha vuelto bastante complejo y de gran escala. Existe una tendencia en la comunidad de investigación para resolver los grandes problemas complejos mediante el uso de modernas técnicas de optimización heurística. Esto se debe principalmente al tiempo de cómputo y la inadecuación de las técnicas clásicas en muchas circunstancias, como son, la gran cantidad de variables y restricciones.

El estudio de (Paiva, 2009) propone un modelo de optimización para soportar las decisiones en un ingenio azucarero, con un modelo que resultó en 12,306 variables, donde 5,796 fueron binarias y 6902 restricciones. En el trabajo de (Zhang, 2012) se construyó un modelo de programación lineal de enteros mixtos que caracteriza un problema de APP con la expansión de capacidad en un sistema de fabricación que incluye múltiples centros de actividad. Utilizaron un método heurístico basado en el cambio de capacidad con relajación lineal para resolver el problema. (Ramezani, 2012) consideraron sistemas de múltiples periodos, múltiples productos y múltiples máquinas con decisiones de configuración, desarrollaron un modelo de programación lineal de enteros mixtos para sistemas APP de dos fases generales e implementaron un algoritmo genético y búsqueda tabú (TS) para resolver el modelo. En el trabajo de (Abu, 2016) se resuelve un problema de APP por medio de Recocido Simulado. En el trabajo de (Mehdizadeh, 2018) propone una solución a través de GA al problema de producción agregada. En el estudio de (Anand, 2017) se aplica Python en la solución de un problema de APP. En el trabajo de (Al-e, 2012) se



**Figura 1: Fuerza de gravitación entre varias partículas; La Fuerza<sub>1</sub> es la resultante sobre la Masa<sub>1</sub>.**

aplica un algoritmo genético para resolver un problema de planeación agregada multiobjetivo.

### 1.2. Algoritmos metaheurísticos

Los algoritmos metaheurísticos imitan los procesos físicos o biológicos que se encuentran en la naturaleza. Algunos de los algoritmos más famosos son los Algoritmos Genéticos (GA), Optimización de Cúmulo de Partículas (PSO) (Eberhart, 1995), Recocido Simulado (SA) (Kirkpatrick, 1983), Sistemas Inmunes Artificiales (AIS) (Farmer, 1986).

Los GA están inspirados en la teoría de la evolución de Darwin, el algoritmo SA se basa en los efectos termodinámicos, los AIS simulan un sistema inmunológico artificial y el PSO simula el comportamiento social y cognitivo de las parvadas de aves o bancos de peces.

## 2. Algoritmo de Búsqueda Gravitacional

El algoritmo de búsqueda gravitacional fue publicado en el trabajo de (Rashedi, 2009), donde los investigadores dieron a conocer la potencialidad y eficacia de esta técnica. Esto se puede ver en el surgimiento de implementaciones utilizando el GSA. Se ha aplicado y mejorado para diferentes estudios. Entre los trabajos destacados en GSA se encuentra en la última revisión de la literatura que el GSA ha sido ampliamente usado para resolver problemas relacionados con el costo de producción y consumo energético (Ibrahim, 2018), (Narimani, 2014), en trabajos de investigación que involucran la solución de problemas relacionados al tiempo de producción y emisiones (Chaturvedi, 2015).

Uno de estos algoritmos inspirado por las leyes físicas es el Algoritmo de Búsqueda Gravitacional, tomando como principio que cada partícula en el universo atrae a cada una de las otras a causa de la gravedad. La Gravitación es la tendencia de las masas a acelerarse hacia las otras. La ley de la gravitación universal de Newton establece que cada partícula atrae a otra con una fuerza que es directamente proporcional a sus masas e inversamente proporcional al cuadrado de la distancia entre ellas.

Para describir el GSA considere un sistema con  $s$  masas como el de la figura 1 en el cual la posición de la  $i$ -ésima masa está definida como sigue:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n), \quad i = 1, 2, \dots, s \quad (1)$$

donde  $x_i^d$  es la posición de la  $i$ -ésima masa en la  $d$ -ésima dimensión y  $n$  es la dimensión del espacio de búsqueda. La masa de cada agente se calcula después de conocer el valor de la función costo (fitness) con las ecuaciones 2 y 3.

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (2)$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^s q_j(t)} \quad (3)$$

donde  $M_i(t)$  y  $fit_i(t)$  representan la masa y el valor fitness del agente  $i$  en  $t$ , respectivamente. Para un problema de minimización,  $worst(t)$  y  $best(t)$  se definen como se indica en las ecuaciones 4 y 5,

$$best(t) = \min_{(j=1\dots s)} fit_j(t) \quad (4)$$

$$worst(t) = \max_{(j=1\dots s)} fit_j(t) \quad (5)$$

Para calcular la aceleración de un agente, deben de considerarse el total de fuerzas de un conjunto de masas más pesadas aplicadas sobre el agente basadas en la ley de la gravitación (ec. 6), la cual es seguida del cálculo de la aceleración del agente usando la ley de movimiento (ec. 7). Posteriormente, se actualiza la posición y velocidad del agente de acuerdo a las ecuaciones 8 y 9,

$$F_i^d(t) = \sum_{j \in k_{best}, j \neq i} rand_j * G(t) \frac{M_j(t)M_i(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (6)$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} = \sum_{j \in k_{best}, j \neq i} rand_j * G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (7)$$

$$v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t) \quad (8)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (9)$$

donde  $rand_i$  y  $rand_j$  son dos números aleatorios uniformemente distribuidos en el intervalo de  $[0,1]$ ,  $\varepsilon$  es un valor muy pequeño que evita que la ecuación se indetermine cuando las partículas están en la misma posición,  $R_{ij}(t)$  es la distancia Euclidiana entre dos agentes  $i$  y  $j$ , que se define como  $\|x_i(t), x_g(t)\|$ ,  $k_{best}$  es el conjunto de los primeros  $k$  agentes con el mejor valor fitness y la mayor masa, la cual está en función del tiempo, inicializada a  $k_0$  al inicio y decreciendo con el tiempo. Aquí  $k_0$  se aplica a  $s$  (número total de agentes) y decrece linealmente a 1. En el GSA, la constante gravitacional,  $G$ , tomará un valor inicial  $G_0$ , y se reducirá con el tiempo como se indica en (Rashedi, 2009):

$$G(t) = G_0 * e^{-\frac{Ct}{T}}, t = 1, 2, \dots, T \quad (10)$$

donde  $G(t)$  es la constante de gravitación en la iteración  $t$ ,  $G_0$  y  $C$  son coeficientes de control del GSA,  $t$  es el número de iteración y  $T$  es el número total de iteraciones (también conocida como tiempo de vida del sistema).

El algoritmo GSA se compone de los siguientes pasos:

1. Identificación del espacio de búsqueda,

2. Inicialización aleatoria,
3. Evaluar fitness de los agentes,
4. Actualizar  $G(t)$ ,  $best(t)$ ,  $worst(t)$  y  $M_i(t)$  para  $i=1,2,\dots,N$ .
5. Calcular el total de fuerzas en todas las direcciones diferentes.
6. Calcular aceleración y velocidad.
7. Actualizar la posición de los agentes.
8. Repetir los pasos 3 al 7 hasta que el criterio de parada se alcance.
9. Fin.

### 2.1. Desempeño del GSA

El desempeño del algoritmo GSA ha sido estudiado en (Rashedi, 2009) con las funciones benchmark del trabajo de (Liang, 2006) que permiten realizar un comparativo en base a los resultados alcanzados. Las funciones benchmark producen una superficie de respuesta de la cual el algoritmo debe encontrar el mínimo. Algunas de las funciones donde se ha encontrado un eficiente desempeño del GSA se muestran en la figura 2.

Estas funciones han sido implementadas en el trabajo de Rashedi para conocer el desempeño de este algoritmo sobre problemas de alto grado de complejidad en cuanto al número de variables y, dado que son funciones matemáticas cuyo valor máximo o mínimo es conocido, es posible saber cuál es la desviación entre el valor alcanzado por el algoritmo y el valor real. Como se puede ver en su trabajo, Rashedi demuestra el gran desempeño para encontrar el mínimo de funciones complejas.

### 3. Planteamiento del problema de APP

El estudio consiste en satisfacer la demanda para 12 meses indicada en la tabla 1. La compañía tiene 20 trabajadores y 1000 unidades fabricadas. Cada trabajador puede producir 1500 unidades por mes. La compañía puede reclutar del mercado laboral local, pero los reclutas deben ser entrenados por un mes por un trabajador antes de que puedan ser empleados para producción. Cada trabajador puede entrenar como máximo a 5 reclutas durante un mes. El pago mensual de un trabajador es \$15,000 por mes cuando es empleado en producción o para entrenamiento. Un trabajador puede ser descansado a un costo de \$5000 mensuales. Despedir un trabajador cuesta \$15000. Cada recluta cobra \$5000 durante su entrenamiento.

El costo por almacenaje de inventario es \$10 por unidad por mes. Cada unidad no entregada a tiempo incurre en una penalización de \$10 por mes hasta que se completa la entrega. Todas las entregas se deben completar en 12 meses. La compañía requiere al final una fuerza de trabajo de 20 trabajadores y 1000 unidades al final del mes 12. El problema de planeación agregada consiste en decidir cuantos despidos, contrataciones, producción, almacenamiento y retrasos deberá realizar la empresa durante el periodo del contrato (Anand, 2017).

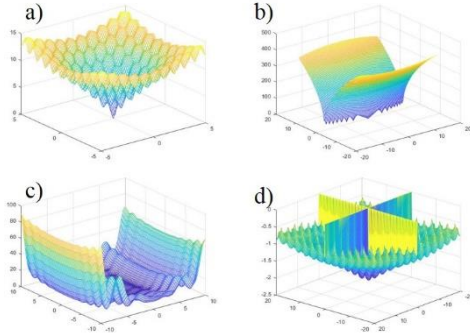


Figura 2: Algunas funciones de prueba utilizadas para realizar la prueba de desempeño de algoritmos de optimización a) Ackley, b) Bukin, c) Levy, d) Crossit.

### 3.1. Modelo matemático

Se consideran las siguientes variables de decisión:

- $W_t$  = Total de trabajadores al inicio del mes  $t$ , antes de despidos.
- $P_t$  = Trabajadores asignados a la producción en el mes  $t$ ,
- $T_t$  = Trabajadores asignados a training en el mes  $t$ ,
- $L_t$  = Trabajadores descansados en el mes  $t$ ,
- $F_t$  = Trabajadores despedidos al inicio del mes  $t$ ,
- $R_t$  = Trabajadores reclutados al inicio del mes  $t$ ,
- $I_t$  = Inventario acumulado al final del mes  $t$ ,
- $S_t$  = Retrasos acumulados al final del mes  $t$ ,
- $X_t$  = Número de unidades producidas durante el mes  $t$ .

Tabla 1: Pronósticos de demanda

Mes (t)	Demanda	Mes (t)	Demanda
1	21306	7	9828
2	20477	8	10273
3	18203	9	14217
4	11106	10	9520
5	5692	11	18007
6	8616	12	21662

La función objetivo representa la suma de los siguientes costos:

- Costo de los trabajadores en producción.
- Costo de los trabajadores en descanso.
- Costo de los trabajadores despedidos
- Costo de los reclutas contratados
- Costo de los trabajadores asignados a training
- Costo por almacenaje de inventario
- Costo por unidades retrasadas

Minimizar:

$$Z = 15000 \sum_{t=1}^{12} P_t + 5000 \sum_{t=1}^{12} L_t + 15000 \sum_{t=1}^{12} F_t + 5000 \sum_{t=1}^{12} R_t + 15000 \sum_{t=1}^{12} T_t + 10 \sum_{t=1}^{12} I_t + 10 \sum_{t=1}^{12} S_t \quad (11)$$

Bajo las siguientes restricciones:

- Tamaño de la fuerza de trabajo

$$W_t = W_{t-1} + R_{t-1} - F_{t-1}, \quad t = 2,3, \dots, 12 \quad (12)$$

La ecuación 12 garantiza que el número total de trabajadores al inicio del mes  $t$  será igual al número al inicio del mes  $t - 1$  mas el número de reclutas en el mes  $t - 1$ , menos el número de despedidos al inicio del mes  $t - 1$ .

- Asignación de la fuerza de trabajo

$$W_t = P_t + T_t + L_t + F_t, \quad t = 2,3, \dots, 12 \quad (13)$$

La ecuación 13 garantiza que el número total de trabajadores al inicio del mes estarán asignados a una de las siguientes actividades: Producción, Entrenamiento de trabajadores reclutados, Descanso, Despedido

- Entrenamiento de trabajadores reclutados

$$R_t \leq 5 * T_t, \quad t = 1,2, \dots, 12 \quad (14)$$

La ecuación 14 garantiza que un trabajador puede entrenar a máximo cinco reclutas.

- Balance entre demanda  $D_t$  e inventario

$$X_t + I_{t-1} = D_t + S_{t-1} + I_t - S_t, \quad t = 1,2, \dots, 12 \quad (15)$$

El lado izquierdo de la ecuación 15 es la suma de la producción actual y el inventario del periodo  $I_{t-1}$ . Por lo que el total debe satisfacer la demanda del mes  $t$ .

- Capacidad de producción

$$X_t \leq 1500 * P_t, \quad t = 1,2, \dots, 12 \quad (16)$$

La ecuación 16 garantiza que cada trabajador puede producir máximo 1500 unidades al mes.

- Restricción de valores positivos

$$P_t, T_t, F_t, R_t, I_t, S_t, X_t \geq 0, \quad t = 1,2, \dots, 12 \quad (17)$$

Tabla 2: Intervalos de búsqueda

Variable	Mínimo	Máximo
$W_t$	0	20
$P_t$	0	20
$L_t$	0	10
$F_t$	0	10
$R_t$	0	10
$T_t$	0	10
$X_t$	6000	25000
$I_t$	0	10
$S_t$	0	10

## 4. Resultados

Se desarrolló una rutina en Matlab tomando como base el algoritmo GSA publicado por Rashedi en el sitio Web de Matlab (Rashedi, 2011). Se adaptó la función objetivo al problema presentado en este trabajo y se ejecutó la rutina empleando los intervalos de búsqueda de la tabla 2, obteniendo los resultados mostrados en las tablas 3 y 4.

El problema fue solucionado de igual forma aplicando el software LINGO con el fin de comparar ambas técnicas, los resultados obtenidos en LINGO se muestran en las tablas 5 y 6. Con estos resultados la función objetivo de la ecuación 11 encontrada por el algoritmo GSA fue de 717,713 mientras que la encontrada por LINGO mediante el método *Branch and Bound* fue de 719,905.

El manejo de restricciones en el GSA se realizó utilizando la técnica de (Yadav, 2013) y (Chehourri, 2016), donde se realiza un recuento de las restricciones no cumplidas y se asigna un factor de peso a éstas. Este término es agregado a la función objetivo y se corre el algoritmo considerando estos términos adicionales. Debido a esto es posible que el GSA pueda encontrar una solución no óptima si no se asignan los pesos adecuados a cada una de las restricciones.

Tabla 3: Solución trabajadores algoritmo GSA

Mes (t)	$W_t$	$P_t$	$T_t$	$L_t$	$F_t$	$R_t$
0	20	0	0	0	0	0
1	20	7	8	8	9	7
2	4	0	7	5	1	1
3	9	6	8	1	3	8
4	11	7	5	4	10	9
5	2	2	7	7	2	6
6	6	1	8	1	2	9
7	15	3	4	10	6	6
8	13	4	4	8	8	5
9	13	11	7	6	6	6
10	14	4	2	5	2	5
11	10	15	10	8	3	1
12	20	0	2	7	2	0

Tabla 4: Solución producción algoritmo GSA

Mes (t)	$I_t$	$S_t$	$X_t$	$D_t$
0	1000	0	0	0
1	7	10	18133	21306
2	1	8	23331	20477
3	8	9	12189	18203
4	9	10	10035	11106
5	6	0	6522	5692
6	9	5	13283	8616
7	6	9	12875	9828
8	5	5	6924	10273
9	6	5	9969	14217
10	5	9	19747	9520
11	1	1	17575	18007
12	1000	0	8705	21662

Tabla 5: Solución trabajadores LINGO®

Mes (t)	$W_t$	$P_t$	$T_t$	$L_t$	$F_t$	$R_t$
0	20	0	0	0	0	0
1	20	14	0	0	6	0
2	14	13	0	0	1	0
3	13	12	0	0	1	0
4	12	8	0	1	3	0
5	9	4	0	5	0	0
6	9	6	0	3	0	0
7	9	7	0	2	0	0

8	9	7	1	0	0	2
9	11	10	0	1	0	0
10	11	6	1	0	0	5
11	16	12	1	0	0	4
12	20	15	0	5	0	0

Tabla 6: Solución producción LINGO®

Mes (t)	$I_t$	$S_t$	$X_t$	$D_t$
0	1000	0	0	0
1	694	0	21000	21306
2	0	283	19500	20477
3	0	486	18000	18203
4	0	0	11592	11106
5	0	0	5692	5692
6	0	0	8616	8616
7	0	0	9828	9828
8	0	0	10273	10273
9	689	0	14906	14217
10	169	0	9000	9520
11	162	0	18000	18007
12	1000	0	22500	21662

### 5. Conclusiones

En la presente investigación se realizó la aplicación del algoritmo de búsqueda y optimización GSA en la solución de un problema de APP, el cual permite encontrar el mínimo de la función objetivo sujeto a las restricciones impuestas. Este estudio no se había realizado antes con esta técnica.

Se debe prestar especial atención al manejo de restricciones ya que se deben asignar pesos adecuados a cada término adicional a considerar como restricción.

Como trabajo futuro se propone realizar el análisis de varianza o el diseño de un experimento factorial que ayude a encontrar los parámetros de desempeño del GSA en este tipo de problemas, así como la aplicación del algoritmo GSA multiobjetivo y de otros para la solución de problemas con múltiples líneas de producto y varias instalaciones que comparten recursos.

### Referencias

Able, B. C., 1945. Nombre del artículo. Nombre de la revista 35, 123–126. DOI: 10.3923/ijbc.2010.190.202

Abu Bakar, M. R., Bakheet, A. J. K., Kamil, F., Kalaf, B. A., Abbas, I. T., Soon, L. L., 2016. Enhanced simulated annealing for solving aggregate production planning. *Mathematical Problems in Engineering* 2016.

Al-e, S. M. J. M., Aryanezhad, M. B., Sadjadi, S. J., et al., 2012. An efficient algorithm to solve a multi-objective robust aggregate production planning in an uncertain environment. *The International Journal of Advanced Manufacturing Technology* 58 (5-8), 765–782.

Anand Jayakumar, A., Krishnaraj, C., Nachimuthu, A., 2017. Aggregate production planning: Mixed strategy. *Pak. J. Biotechnol.* Vol 14 (3), 487–490.

Bu\_a, E., Taubert, W., 1972. Production-inventory systems planning and control. Tech. rep.

Chaturvedi, N. D., Bandyopadhyay, S., 2015. Targeting aggregate production planning for an energy supply chain. *Industrial & Engineering Chemistry Research* 54 (27), 6941–6949.

Chehourri, A., Younes, R., Perron, J., Ilinca, A., 2016. A constraint-handling technique for genetic algorithms using a violation factor. arXiv preprint arXiv:1610.00976.

- Cheraghalikhani, A., Khoshalhan, F., Mokhtari, H., 2019. Aggregate production planning: A literature review and future research directions. *International Journal of Industrial Engineering Computations* 10 (2), 309–330.
- da Silva, C. G., Figueira, J., Lisboa, J., Barman, S., 2006. An interactive decision support system for an aggregate production planning model based on multiple criteria mixed integer linear programming. *Omega* 34 (2), 167–177.
- Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on.* IEEE, pp. 39–43.
- Farmer, J. D., Packard, N. H., Perelson, A. S., 1986. The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena* 22 (1–3), 187–204.
- Holt, C. C., Modigliani, F., Muth, J. F., 1956. Derivation of a linear decision rule for production and employment. *Management Science* 2 (2), 159–177.
- Holt, C. C., Modigliani, F., Simon, H. A., 1955. A linear decision rule for production and employment scheduling. *Management Science* 2 (1), 1–30.
- Ibrahim, A. M., Swief, R. A., 2018. Comparison of modern heuristic algorithms for loss reduction in power distribution network equipped with renewable energy resources. *Ain Shams Engineering Journal*.
- Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1983. Optimization by simulated annealing. *science* 220 (4598), 671–680.
- Liang, J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., Coello, C. C., Deb, K., 2006. Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics* 41 (8), 8–31.
- Narimani, M. R., Vahed, A. A., Azizpanah-Abarghoee, R., Javidsharifi, M., 2014. Enhanced gravitational search algorithm for multi-objective distribution feeder reconfiguration considering reliability, loss and operational cost. *IET Generation, Transmission & Distribution* 8 (1), 55–69.
- Mehdizadeh, E., Niaki, S. T. A., Hemati, M., 2018. A bi-objective aggregate production planning problem with learning effect and machine deterioration: Modeling and solution. *Computers & Operations Research* 91, 21–36.
- Nam, S.-j., Logendran, R., 1992. Aggregate production planning—a survey of models and methodologies. *European Journal of Operational Research* 61 (3), 255–272.
- Paiva, R. P., Morabito, R., 2009. An optimization model for the aggregate production planning of a brazilian sugar and ethanol milling company. *Annals of Operations Research* 169 (1), 117.
- Ramezani, R., Rahmani, D., Barzinpour, F., 2012. An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search. *Expert Systems with Applications* 39 (1), 1256–1263.
- Rashedi, E., 2011. Gravitational search algorithm (gsa). [urlhttps://la.mathworks.com/matlabcentral/fileexchange/27756-gravitational-search-algorithm-gsa](https://la.mathworks.com/matlabcentral/fileexchange/27756-gravitational-search-algorithm-gsa).
- Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. Gsa: a gravitational search algorithm. *Information sciences* 179 (13), 2232–2248.
- Yadav, A., Deep, K., 2013. Constrained optimization using gravitational search algorithm. *National Academy Science Letters* 36 (5), 527–534.
- Zhang, R., Zhang, L., Xiao, Y., Kaku, I., 2012. The activity-based aggregate production planning with capacity expansion in manufacturing systems. *Computers & Industrial Engineering* 62 (2), 491–503.