

Trabajo Práctico N°2

1- “A block-based genetic algorithm for disassembly sequence planning”. Algoritmo genético basado en bloques para planeamiento de secuencias de desensamblado.

Dominio de la aplicación:

El planeamiento de la secuencia de desensamblado (DSP) se refiere al estudio del orden secuencial de desensamblado basado en los atributos de las partes. Se busca obtener un método científico de evaluación que pueda contribuir a la determinar el desempeño del diseño de un producto.

Notes	Definition
N	number of parts in a product
J	index for component
seq	index for disassembly sequence
$P_{j,seq}$	penalty function (disassembly component j in sequence seq)
$D_{j,seq}$	score in the change of disassembly direction (disassembly component j in sequence seq)
$T_{j,seq}$	the score in the change of disassembly method or tool (disassembling component j in sequence seq)
MS	the fitness function value
M	the number of chromosomes
m	the amount of mutation
<i>Block-size</i>	the size of a block

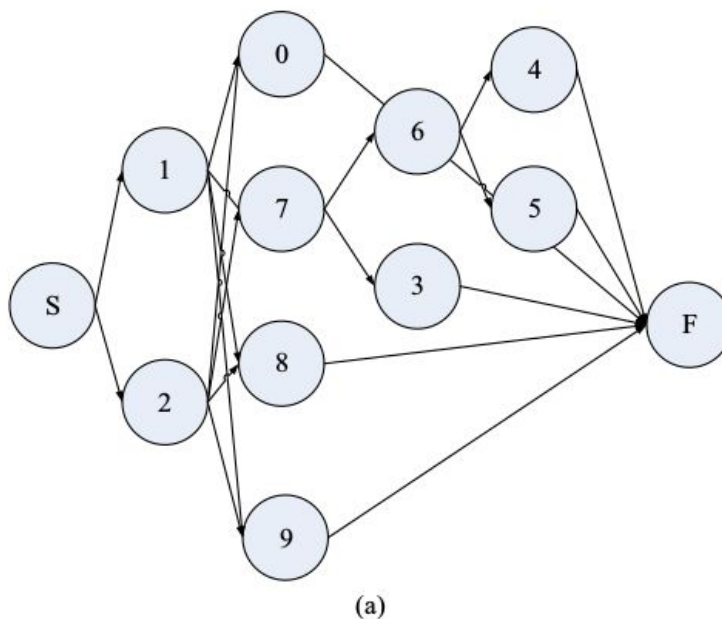
Se tendrán seis direcciones de desensamblado $\pm X$, $\pm Y$, $\pm Z$. Para cambio de 90° se dará un puntaje de 1, mientras que para un cambio de 180° se dará un puntaje de 2. Un puntaje de 1 se dará si se produce algún cambio de herramienta. Estos puntajes se muestran en la siguiente tabla.

Engineering information for K&G example (a) change of disassembly direction (b) change of disassembly tools.

(a)						
Front/Rear	+X	-X	+Y	-Y	+Z	-Z
+X	0	2	1	1	1	1
-X	2	0	1	1	1	1
+Y	1	1	0	2	1	1
-Y	1	1	2	0	1	1
+Z	1	1	1	1	0	2
-Z	1	1	1	1	2	0

(b)				
Front/Rear	T1	T2	T3	T4
T1	0	1	1	1
T2	1	0	1	1
T3	1	1	0	1
T4	1	1	1	0

Un ejemplo utilizando 10 partes se muestra a continuación. En la imagen se muestra un grafo que representa los distintos pasos que deben realizarse para lograr el desensamblado del artefacto a analizar. En la tabla inferior se muestra la herramienta utilizada en cada paso y su dirección.



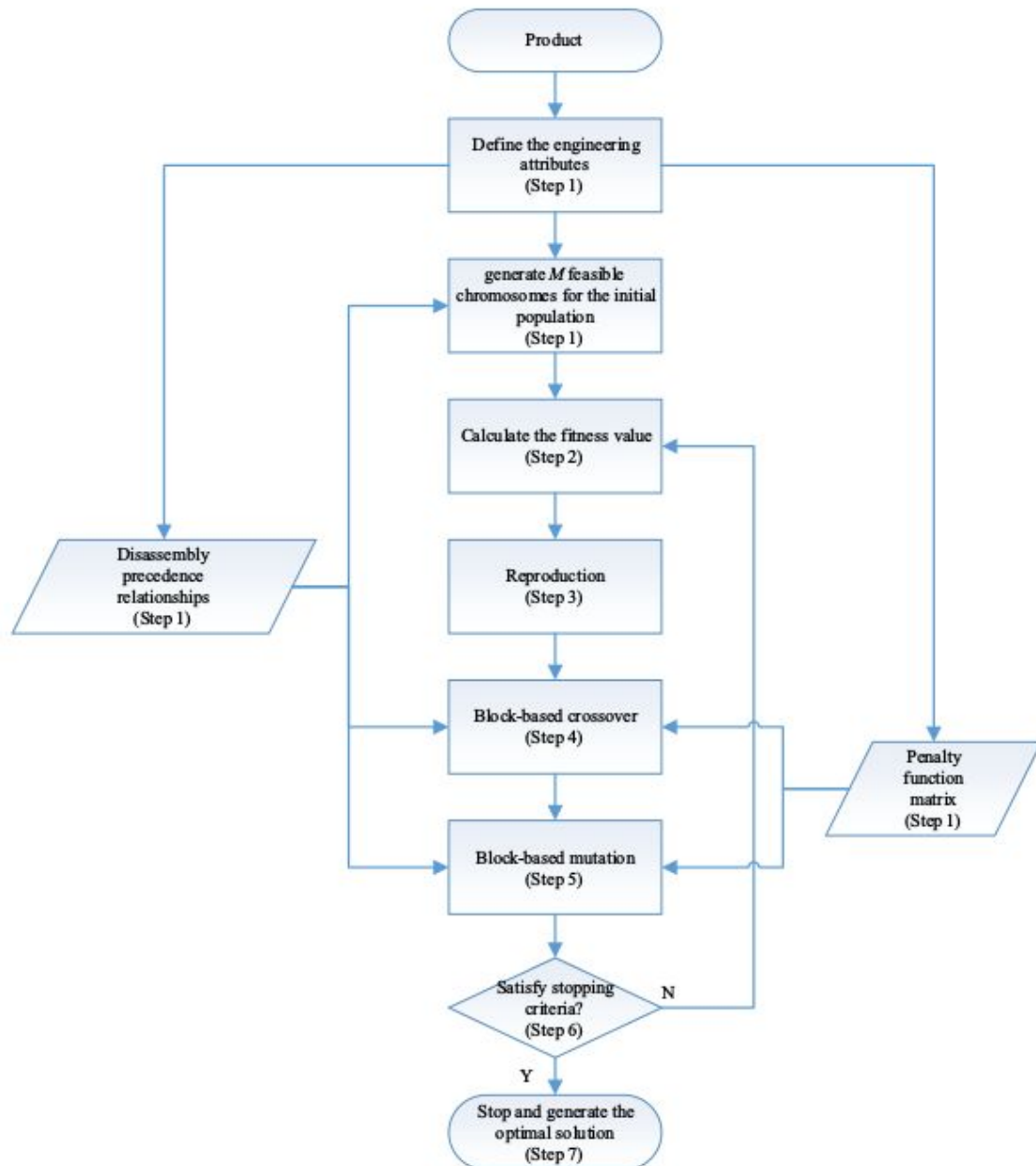
Parts	0	1	2	3	4	5	6	7	8	9
Direction	+Y	+X	+X	+Z	-Z	-Z	-Y	-X	-Z	-Y
Tool	T ₁	T ₂	T ₂	T ₁	T ₂	T ₂	T ₁	T ₁	T ₂	T ₁

La función de penalidad estará expresada por tanto como $P_{j,seq} = D_{j,seq} + T_{j,seq}$.

La función de fitness estará expresada como la suma de las funciones de penalidad entre nodos consecutivos.

$$MS = \sum_{seq=0}^{n-2} P_{j,seq}, \forall j, j = 0, \dots, n-1$$

Algoritmo genético basado en bloques: La diferencia principal entre éste algoritmo y el K&G radica en la introducción de la matriz de penalidad en el mecanismo de crossover y de mutación.



Los pasos del algoritmo se detallarán a continuación.

Paso 1: Los cromosomas son codificados en dígitos decimales para definir los atributos de las partes. Se generan M cromosomas factibles como padres iniciales.

Paso 2: Se calcula el valor de la función de fitness para cada cromosoma para su evaluación.

Paso 3: Por método de la ruleta se reproducen un número igual de cromosomas para la próxima iteración.

Paso 4: Mecanismo de crossover basado en bloques.

Paso 5: Mecanismo de mutación basado en bloques.

Paso 6: Verificar si ha terminado el proceso de evolución, el máximo número de generaciones sirve como condición de parada. En caso de no satisfacer la condición se procede con el paso 2.

Paso 7: El algoritmo termina generando un orden de secuencia óptimo o cercano al óptimo.

Mecanismo de crossover basado en bloques: Consiste en dos fases

Fase 1: Un bloque en el cromosoma se mantiene luego del crossover, el bloque con mejores genes se mantiene luego del crossover. El tamaño del bloque puede ser del 0%-100% del cromosoma y se genera aleatoriamente. El algoritmo será el siguiente:

Paso 4.1: Se seleccionan dos cromosomas aleatoriamente.

Paso 4.2: Se divide cada cromosoma en $(n - \text{Tamaño-bloque} + 1)$ bloques.

Paso 4.3: Se calcula el valor de la función de fitness para cada bloque.

Paso 4.4: El bloque con mejor fitness es el bloque óptimo. Se selecciona este bloque.

Fase 2: Los códigos genéticos de los cromosomas se intercambian en la nueva generación.

Paso 4.5: Se crea una descendencia-2 del padre-1 y una descendencia-1 del padre-2. Se colocan los bloques correspondientes de los padres en sus mismas posiciones de sus respectivos descendientes.

Paso 4.6: Buscar por códigos genéticos repetidos y faltantes en la descendencia.

Paso 4.7: Eliminar el código genético repetido en la descendencia.

Paso 4.8: Buscar por todas las posiciones posibles para insertar los códigos genéticos faltantes.

Paso 4.9: Calcular los valores de fitness para todos los posibles puntos de inserción.

Paso 4.10: Seleccionar la mejor posición.

Paso 4.11: Insertar los códigos genéticos en la descendencia.

Mecanismo de mutación basado en bloques:

Paso 5.1: Seleccionar aleatoriamente un punto del cromosoma y eliminarlo temporalmente del cromosoma.

Paso 5.2: Encontrar el rango de inserción del punto de mutación, confirmando que las soluciones generadas son factibles.

Paso 5.3: Calcular las funciones de fitness para todos los puntos de mutación en el rango posible.

Paso 5.4: Encontrar el mejor punto de inserción e insertar el gen.

Paso 5.5: Repetir el procedimiento m veces y se finaliza el proceso de mutación.

Step 5.4: Find the best fitness function value for the insertion of the mutation point.

Step 5.5: Repeat Step 5.1–Step 5.4 m times (m is the mutation amount) and finish the mutation mechanism.

Resultados:

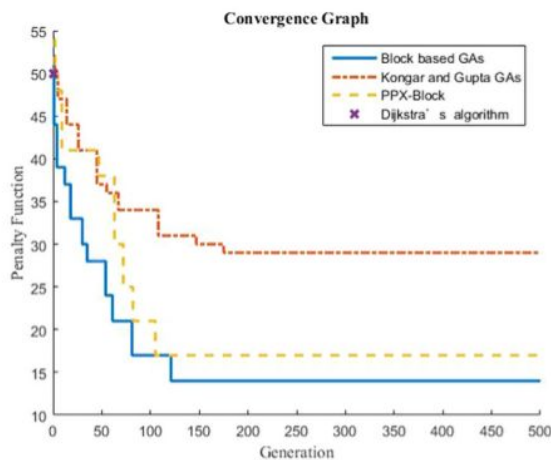


Fig. 10. The convergence graph of Printer.

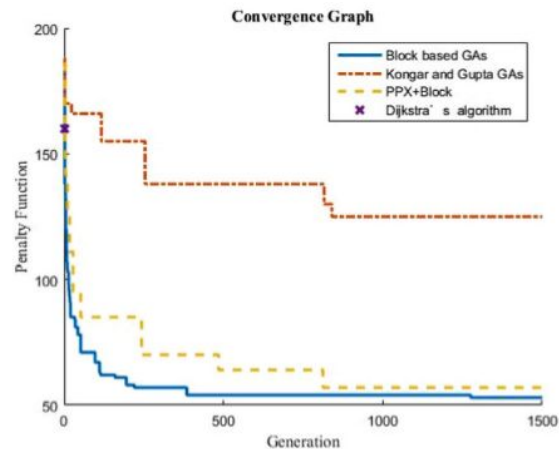


Fig. 12. The convergence graph for the virtual 150 nodes example.

Los algoritmos de Dijkstra no mejoran la solución a medida que el programa progresa, pero debido a su pequeño tiempo de ejecución pueden ser considerado un método apropiado cuando no se requiere una solución óptima.

El algoritmo genético basado en bloques mejora la búsqueda de una solución. Comparado con los algoritmos genéticos de K&G, el algoritmo genético basado en bloque es más eficiente en mejorar la calidad de solución mientras la complejidad del problema aumenta.

2- Mining Internet of Things for intelligent objects using genetic algorithm. Minando internet de las cosas en búsqueda de objetos inteligentes usando algoritmos genéticos.

Dominio de la aplicación:

El internet de las cosas está sobrepoblado con un gran número de objetos y millones de servicios e interacciones. Por tanto la capacidad de buscar el objeto adecuado que provea el servicio adecuado es importante. La unión de las IoT y las redes sociales, Social Internet of Things (SioT) lo ha hecho posible. La idea principal de SioT es que cada objeto en el IoT pueda usar sus relaciones para buscar un servicio específico.

El modelo de interacción tradicional de base en la idea de que los humanos están buscando información. Sin embargo en el IoT el modelo debe cambiar hacia la interacción objeto a objeto lo que significa que el objeto buscará un servicio en otros objetos. Si bien muchos modelos se han propuesto, estos se basan en un modelo centralizado. Éstos poseen la desventaja de que no pueden escalarse adecuadamente. Para sobrellevar esta limitación se propone un acercamiento basado en SioT.

En el enfoque basado en SioT los objetos tendrán la capacidad de buscar el servicio deseado utilizando objetos "amigos" a través de conexiones posibles entre ellos. Cada nodo tendrá una gran cantidad de nodos manejar lo que influirá negativamente en su tiempo de búsqueda. Por lo que es recomendable limitar su espacio de búsqueda.

En IoT cada nodo es un objeto que puede establecer una relación con otros objetos según reglas definidas por usuario.

Parent-object relationship (POR).

Co-location object relationship (CLOR).

Co-work object relationship (CWOR).

Owner-object relationship (OOR).

Social-object relationship (SOR).

El paper busca solucionar el problema unir nodos. Se evalúan cinco formas diferentes de estrategias de unión usando simulación y posteriormente se propone una solución utilizando un algoritmo genético.

El desempeño se evalúa teniendo en cuenta los siguientes parámetros.

- 1- Grado promedio de conexión: mide el número promedio de relaciones de “amistad” de cada nodo.
- 2- Longitud de camino promedio: Mide el tiempo de conexión promedio entre dos nodos de la red.
- 3- Coeficientes del cluster local: mide la propagación de los nodos y las conexiones en la red.

La meta es ayudar a los nodos a seleccionar las mejores relaciones para mejorar el desempeño de la red. Al inicio un nodo aceptará todas las solicitudes de relación hasta alcanzar el máximo número de nodos. Luego el nodo usará las siguientes estrategias para manejar las nuevas solicitudes.

Primera estrategia: rechazar nuevas solicitudes al alcanzar un máximo N de relaciones.

Segunda estrategia: Cada nodo ordena la lista de relaciones en orden decreciente de grado de conexión y acepta los N mejores relaciones.

Tercera estrategia: Cada nodo ordena la lista de en orden creciente de grado de conexión y acepta las N mejores relaciones.

Cuarta estrategia: Luego de recibir una nueva conexión, cada nodo ordena a sus relaciones en orden decreciente de cantidad de amigos comunes y acepta los mejores N relaciones.

Quinta estrategia: Similar a la cuarta, pero en orden creciente.

En la siguiente figura se muestra como se vería el grafo de las conexiones entre nodos

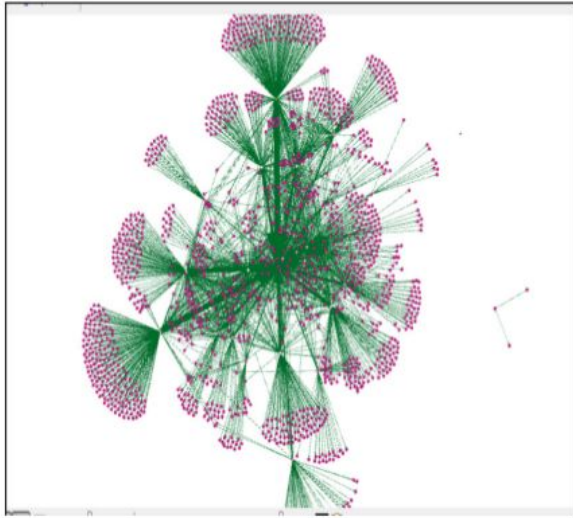
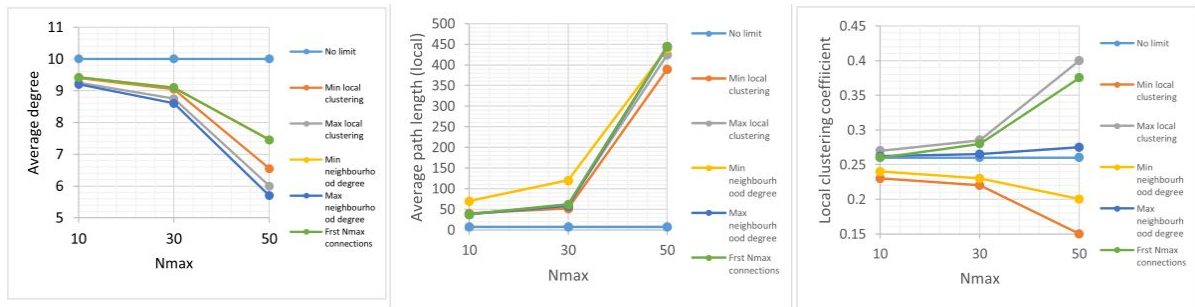
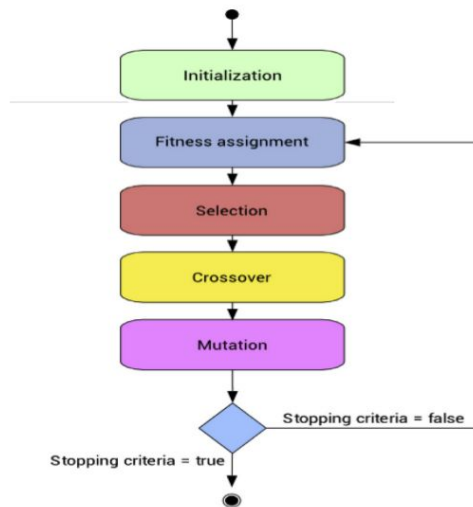


Fig. 1. The main graph.



Las siguientes figuras muestran los resultados promedio para las distintas estrategias repetidas 10 veces para las tres medidas de desempeño.

El algoritmo genético implementado, responde al siguiente esquema



Los parámetros del algoritmo genético son los siguientes:

1. Tamaño de la población: 1000 nodos.
2. Función de fitness: La siguiente formula fue utilizada para conseguir el máximo numero de relaciones.

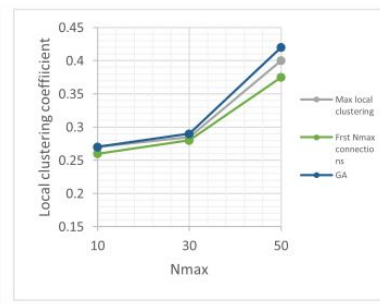
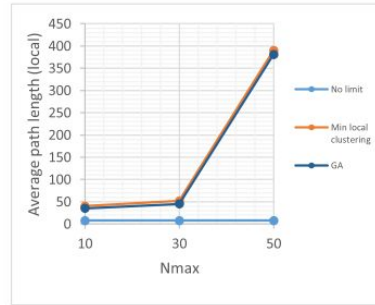
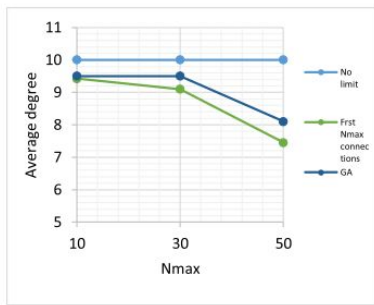
$$F(x) = \frac{\text{Number of Common Friends}}{\text{Number of Friends}_S \times \text{Number of Friends}_R}$$

Donde “Number of Friends S” es el número de relaciones del nodo y “Number of Friends R” es el numero de relaciones del nodo receptor.

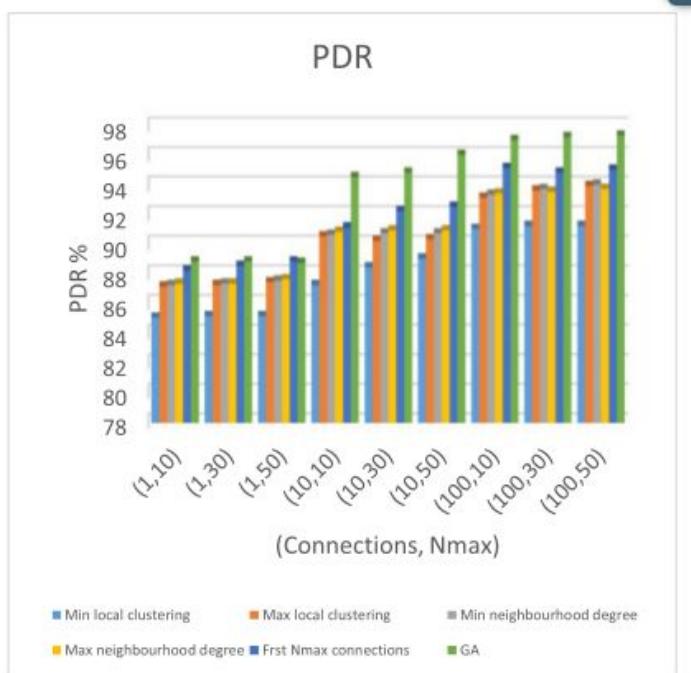
- 3- Gen: Representa un dígito el nodo.
- 4- Cromosoma: representa al nodo
- 5- Crossover: Se selecciona dos cromosomas en función de su valor de fitness, se crea un
- 6- Mutación: Se cambia un gen aleatoriamente.
- 7- G: el proceso se repite 10 veces.

Resultados obtenidos

Las figuras comparan los mejores resultados de la estrategias comparado con el algoritmo genético



En la siguiente figura, se mide el Packet Delivery Ratio (PDR) a partir de 10 ejecuciones por lo que se muestra la eficiencia del algoritmo genético frente a los otro algoritmos utilizados.

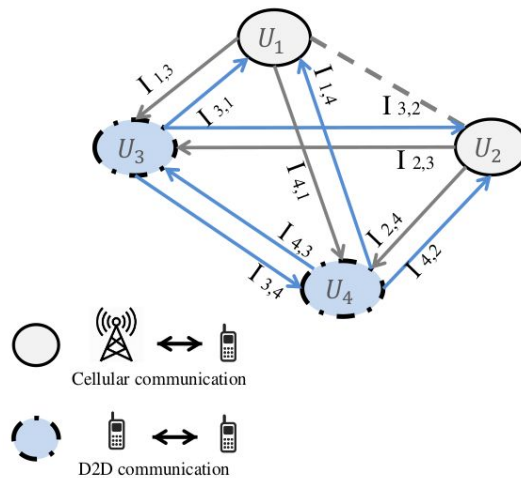


3- “Joint Optimization of Device to Device Resource and Power Allocation based on Genetic Algorithm”. Optimización conjunta de recursos de dispositivo a dispositivo y asignación de potencia basada en algoritmo genético

Dominio de la aplicación: La comunicación Dispositivo a Dispositivo (“D2D” por sus siglas en inglés) es muy importante en los sistemas de comunicación, ya que disminuye la latencia y expande la cobertura de una célula en una red celular. Sin embargo, crea interferencia en las comunicaciones celulares. Se propone un método basado en algoritmos genéticos para minimizar la interferencia y maximizar la eficiencia espectral.

Modelo desarrollado:

Se considera una estación base y una cantidad de usuarios distribuidos aleatoriamente alrededor de la estación base. Los usuarios se clasifican en un grupo de usuarios celulares (que se comunican a través de la estación base) y usuarios de D2D (el usuario se comunica directamente con otro cercano).



El gráfico muestra la interferencia entre cada par de comunicaciones. Los nodos en este gráfico representan una comunicación, ya sea una comunicación celular o D2D. Los bordes son interferencia entre dos nodos conectados. I_{ij} es el peso del borde que indica el conjunto de valores de interferencia desde el nodo i -ésimo al nodo j -ésimo en diferentes bloques de recursos que se define como: $I_{ij} = \{I_{ij,1}, I_{ij,2}, \dots, I_{ij,Q}\}$ donde $I_{ij,r}$ es la interferencia desde el i -ésimo nodo al j -ésimo nodo en el r -ésimo bloque de recursos. La línea punteada entre U_1 y U_2 muestra que no se permite compartir un bloque de recursos entre dos usuarios celulares. Como su receptor es la estación base, si comparten un bloque de recursos, el receptor no puede diferenciar las señales. Por lo tanto, no hay interferencia entre los usuarios celulares.

La eficiencia espectral máxima de la comunicación i -ésima en el r -ésimo bloque de recursos, ya sea de usuario celular o el par D2D, se denota por: $Se_{i,r} = \log_2(1 + SINR_{i,r})$ donde $SINR_{i,r}$ es la señal a la interferencia más la proporción de ruido. El objetivo es maximizar la eficiencia espectral de la red que depende de $SINR_{i,r}$.

Además de la asignación de recursos que determina los usuarios que interfieren, la potencia de transmisión de los usuarios determina la cantidad de interferencia y afecta al SINR y, en consecuencia, a la eficiencia espectral. Por lo tanto, la optimización de la potencia de transmisión de los usuarios ayuda a maximizar la eficiencia espectral de la red y la gestión

de la interferencia. También es evidente que tanto los usuarios celulares como los usuarios de D2D tienen un límite superior para su potencia de transmisión. Los usuarios de celulares también deben cumplir con un SINR mínimo para evitar una reducción severa en la eficiencia espectral mientras comparten su bloque de recursos.

Es entonces un problema de optimización con más de una restricción y para ello se desarrolla el siguiente algoritmo genético.

Individuos: cada individuo presenta la potencia de transmisión y los bloques de recursos dedicados a las comunicaciones.

RB1	{	U_2	U_3	U_4
		p_2	p_3	p_4
RB2	{	U_1	U_5	
		p_1	p_5	

En la figura se ve un ejemplo de individuo, donde dos bloques de recursos están dedicados a cinco comunicaciones ($V=2$ usuarios de celular y $K=3$ pares D2D) cada una con su potencia de transmisión.

Función fitness: es igual a la eficiencia espectral de la red con respecto al cromosoma. La potencia de los usuarios también se optimiza durante el cálculo del valor de fitness. Para este propósito, en cada iteración, se selecciona aleatoriamente una comunicación D2D en cada bloque de recursos (si hay alguna). La potencia de la comunicación seleccionada se optimiza para maximizar la eficiencia espectral del bloque de recursos deseado. Además, el SINR del usuario celular del bloque de recursos (si hay alguno) se establece en SINR mínimo para atender a más usuarios de D2D. Además, si una única comunicación utiliza un bloque de recursos, la potencia de transmisión de la comunicación se establece en su valor máximo.

Selección de padres: La probabilidad de selección de un cromosoma se basa en su valor de aptitud, mientras mayor sea su fitness (en términos de eficiencia espectral) mayor chances tiene de ser elegido como padre.

Crossover: Combina dos padres y produce descendientes para la próxima generación. Al principio, se dedica una longitud aleatoria a cada bloque de recursos de la descendencia, donde la suma de las longitudes debe ser igual al número total de comunicaciones. Esta longitud muestra la cantidad de comunicaciones potenciales que se pueden admitir en el bloque de recursos previsto. Además, se selecciona un número aleatorio en el rango de $[1, V + K]$ que se llama punto de cruce. Determina la cantidad de comunicaciones que se toman de cada padre (por ejemplo, punto de cruce=3 significa tres comunicaciones y sus potencias de comunicación se toman del padre uno y las otras del padre 2).

Mutación: Con el fin de escapar de las soluciones óptimas locales, el algoritmo genético agrega un cambio aleatorio a los cromosomas con una baja probabilidad. Para aplicar un cambio aleatorio en un cromosoma, dos usuarios se seleccionan aleatoriamente y se intercambian. Si el primer usuario es un de un par D2D, el segundo usuario también debe ser de un par D2D.

Probabilidad de mutación: Es baja (0,2) para evitar la búsqueda aleatoria.

Resultados: En todas las simulaciones, se asume que el radio de la red celular es de 500m y hay una estación base en el centro de la red. Los usuarios se distribuyen aleatoriamente. La máxima distancia entre pares de D2D es de 10m. Se prueba la eficacia del algoritmo en distintos escenarios: A) evaluación de la eficiencia espectral de la red, B) evaluación de potencia de interferencia de la red.

- A) En el algoritmo propuesto, a medida que aumenta el número de comunicaciones, también aumenta la eficiencia espectral. La razón radica en tener un número flexible de pares D2D en un bloque de recursos. Se muestra que incluso la asignación aleatoria de recursos puede funcionar mejor que GAAM y la asignación de recursos ortogonales en un entorno denso y lleno de gente. ($Q = 8$ (número de bloques de recursos), $V = 4$ and K varía de 2 a 18)

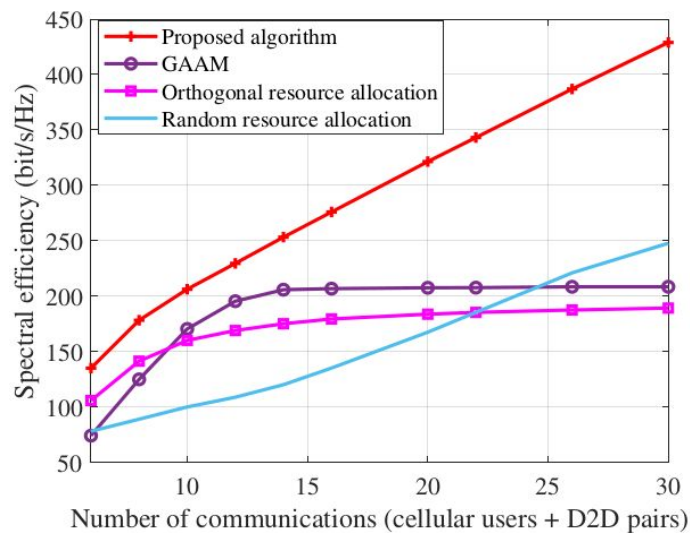


Figure 5: Spectral efficiency for different number of resource blocks.

- B) Se muestra que para un mayor número de repeticiones, hay menos interferencia. En otras palabras, en cada iteración se encuentra una solución mejor y más optimizada.

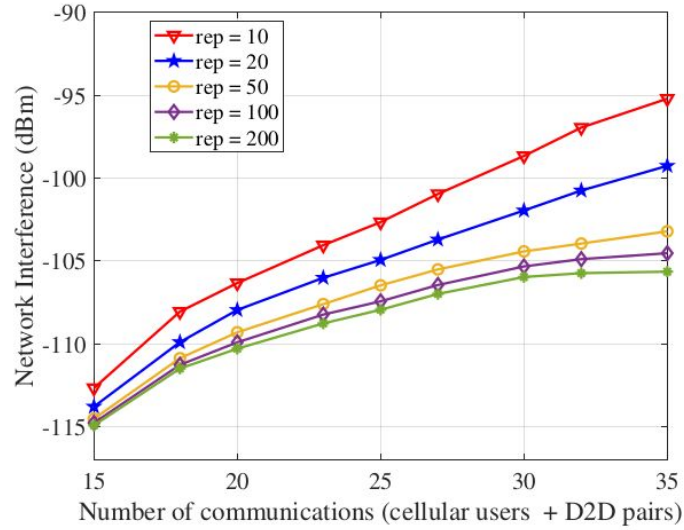


Figure 9: Network interference for different number of repetitions

4- “Genetic algorithm optimization based nonlinear ship maneuvering control”. Control de maniobra no lineal basado en optimización algorítmica genética.

Dominio de la aplicación: El objetivo es maniobrar automáticamente buques de superficie en un camino objetivo determinado. El modelo de maniobra de la nave no es lineal. El algoritmo genético se usa para obtener el ángulo de timón óptimo y la longitud de la trayectoria del objetivo en un paso de tiempo de simulación particular con el objetivo de minimizar el error cuadrático medio de la trayectoria real tomada por la nave con respecto a la trayectoria objetivo.

Modelo desarrollado:

El sistema de coordenadas (X-Z) tiene su origen O en el centro del barco y en la línea central en la superficie libre. El eje X positivo es hacia adelante, el eje Y positivo es hacia estribor, de modo que según la regla del tornillo derecho, el eje Z positivo es verticalmente hacia abajo y cualquier rotación hacia estribor es positiva.

$$\Delta \dot{u} = \frac{\Delta X_F}{m - X_{\dot{u}}}$$

$$\Delta \dot{v} = \frac{(I_z - N_{\dot{r}})\Delta Y_F - (mX_G - Y_{\dot{r}})\Delta N}{(m - Y_{\dot{v}})(I_z - N_{\dot{r}}) - (mX_G - Y_{\dot{r}})(mX_G - N_{\dot{v}})}$$

$$\Delta \dot{r} = \frac{(m - Y_{\dot{v}})\Delta N - (mX_G - N_{\dot{v}})\Delta Y_F}{(m - Y_{\dot{v}})(I_z - N_{\dot{r}}) - (mX_G - Y_{\dot{r}})(mX_G - N_{\dot{v}})}$$

Donde m es la masa del barco, I_z es el momento de inercia de la nave en el eje de oscilación en el eje vertical, u es la velocidad en el eje x, v es la velocidad en el eje y, r es la proporción de oscilación en el eje vertical, X_G es la coordenada x del centro de gravedad del barco, Δu es una pequeña perturbación al valor nominal de u, Δv es una pequeña

perturbación al valor nominal de v , Δr es una pequeña perturbación al valor nominal de r y ΔN es una perturbación en el momento de oscilación en el eje vertical.

Además: $(m - X_{ii}) \Delta \dot{u} = \Delta X_F$

$(m - Y_{\dot{v}}) \Delta \dot{v} + (mX_G - Y_{\dot{r}}) \Delta \dot{r} = \Delta Y_F$

$(mX_G - N_{\dot{v}}) \Delta \dot{v} + (I_Z - N_{\dot{r}}) \Delta \dot{r} = \Delta N$

Individuos: serán p configuraciones $C(j)$ dadas por el parámetro $v_i = (i=1,2)*p..$ Tales configuraciones $C(j)$ ($j=1,2,...,p$) donde p es par, se crean usando: $v_{ij} = v_{Li} + \alpha(v_{Ui} - v_{Li})$ donde α es un número aleatorio entre 0 y 1 que se genera cada vez que se utiliza la ecuación previamente presentada. Es decir, que la aplicación de esta ecuación requiere una generación de $2p$ números aleatorios α .

Función fitness: se utiliza "tournament selection" o selección por torneos, que consiste en elegir k números entre 1 y p , y después elegir el mínimo de esos k números. Se repite p veces para obtener un vector de selección T .

Selección de padres: Se calcula el error cuadrático medio respecto de la trayectoria deseada para cada una de las configuraciones y se ordenan de forma ascendente de acuerdo a sus valores. El conjunto ordenado de p configuraciones al final de este paso es la población de padres. En este trabajo, $k=4$. Entonces se generan 4 números aleatorios entre 0 y 1 y se usan para calcular $q_k = \text{ceil}(p\beta_k)$ donde $\text{ceil}(x)$ establece x al entero más cercano hacia el infinito.

Crossover: Se genera un número aleatorio α' entre 0 y 1. Si $\alpha' \leq \alpha$ (donde $\alpha = 0,8$), entonces se generan dos números aleatorios α_1 y α_2 (ambos entre 0 y 1) y se obtienen dos enteros $n_1 = \text{ceil}(p\alpha_1)$ y $n_2 = \text{ceil}(p\alpha_2)$. Del vector de selección $T_{m(=1 a p)}$ se obtiene $r_1 = T_{n_1}$ y $r_2 = T_{n_2}$. Si $r_1 = r_2$ se elige otro α_2 hasta que sean distintos. Luego se generan N números aleatorios (N : número de parámetros de cada configuración) $\gamma_i (i = 1, 2, ..., N)$ entre 0 y 1.

$$\Delta i = (2\gamma_i)^{\eta+1} \quad (\text{si } \gamma_i < 0.5)$$

$$\Delta i = 1/[2(1 - \gamma_i)]^{\eta+1} \quad (\text{si } \gamma_i \geq 0.5)$$

Luego se generan dos nuevas configuraciones usando:

$$d = d + 1; \quad \overline{C}^{(d)} : v_i^{(d)} = \frac{1}{2} \{ (1 - \Delta i) v_i^{(r_1)} + (1 + \Delta i) v_i^{(r_2)} \}; \text{ hasta } d=p$$

$$d = d + 1; \quad \overline{C}^{(d)} : v_i^{(d)} = \frac{1}{2} \{ (1 + \Delta i) v_i^{(r_1)} + (1 - \Delta i) v_i^{(r_2)} \}; \text{ hasta } d=p$$

Si el parámetro calculado es menor al límite inferior, se iguala a ese límite. Si es mayor al límite superior, se iguala a ese límite.

Mutación: Si $\alpha' > \alpha$, genera un número aleatorio α_1 entre 0 y 1 y obtiene el integral $n_1 = \text{ceil}(p\alpha_1)$ y utilizando el vector de selección T se obtiene $r_1 = T_{n_1}$. Se generan N números aleatorios (N : número de parámetros de cada configuración) $\gamma_i (i = 1, 2, ..., N)$ entre 0 y 1.

$$\Delta i = (2\gamma_i)^{\frac{1}{\eta+1}} - 1 \quad (\text{si } \gamma_i < 0.5)$$

$$\Delta i = 1 - [2(1 - \gamma_i)]^{\frac{1}{\eta+1}} \quad (\text{si } \gamma_i \geq 0.5)$$

Luego se genera una nueva configuración usando:

$$d = d + 1; \quad \overline{C}^{(d)} : v_i^{(d)} = v_i^{(r_1)} + \Delta i (v_{Ui} - v_{Li}); \text{ hasta } d=p$$

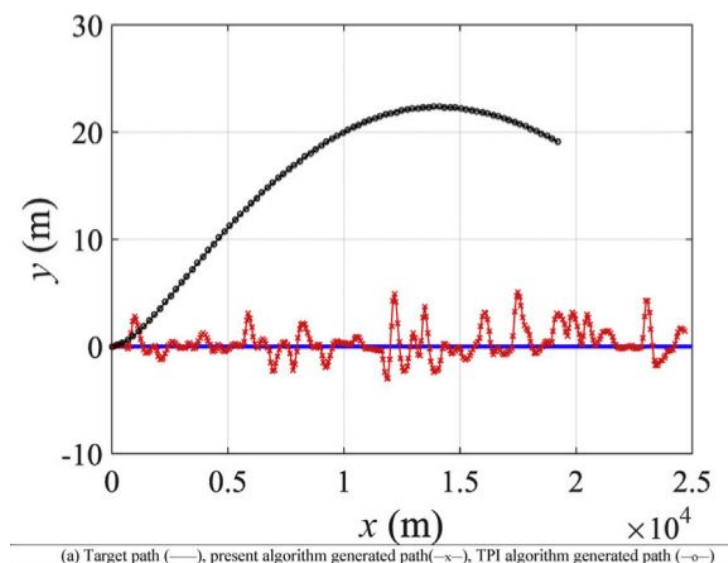
Si el parámetro calculado es menor al límite inferior, se iguala a ese límite. Si es mayor al límite superior, se iguala a ese límite.

Criterio de parada: Se decide parar luego de 20 generaciones, sin otro criterio de convergencia. Se observó que el algoritmo desarrollado tiene bajo rendimiento para menos de 20 iteraciones y para más iteraciones aumenta mucho el tiempo de ejecución sin mejoras significativas en los resultados. La mejor configuración es un conjunto de dos valores: λ, δ que se interpreta como “maniobrar el barco una distancia igual a $\lambda * L$ (L: longitud del barco) mientras se mantiene el timón a un ángulo constante de valor δ .

Resultados:

A) Control sobre una línea recta

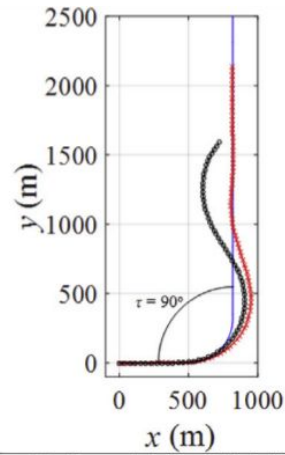
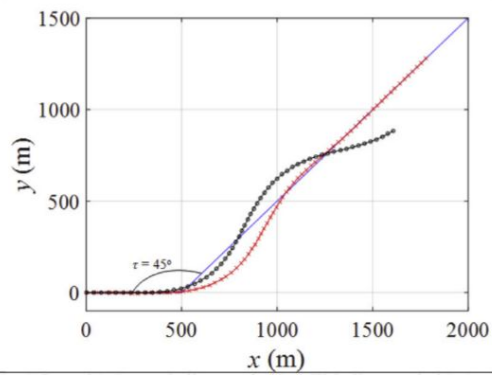
Se ha considerado un camino de línea recta de longitud de 25,000 m. Comparación de las rutas:



Es evidente a partir de que aunque el control sobre una línea recta es oscilatorio en ambos casos, los errores de compensación normalizados son mucho menores con el controlador actual basado en GA (3%) en comparación con el algoritmo TPI (15%).

B) Control sobre camino que consiste en intersección de líneas rectas

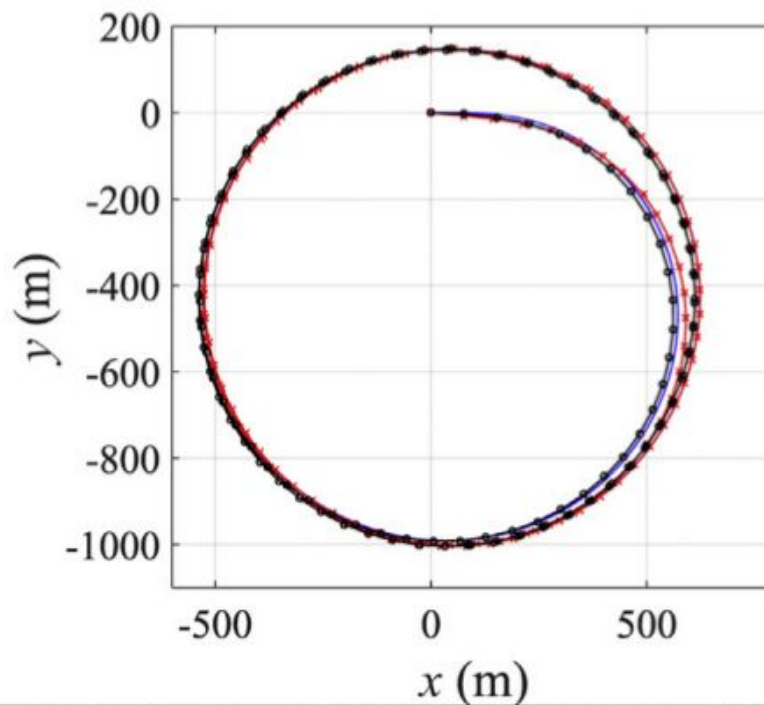
Se toma como camino deseado dos líneas rectas que se cruzan en diferentes ángulos:



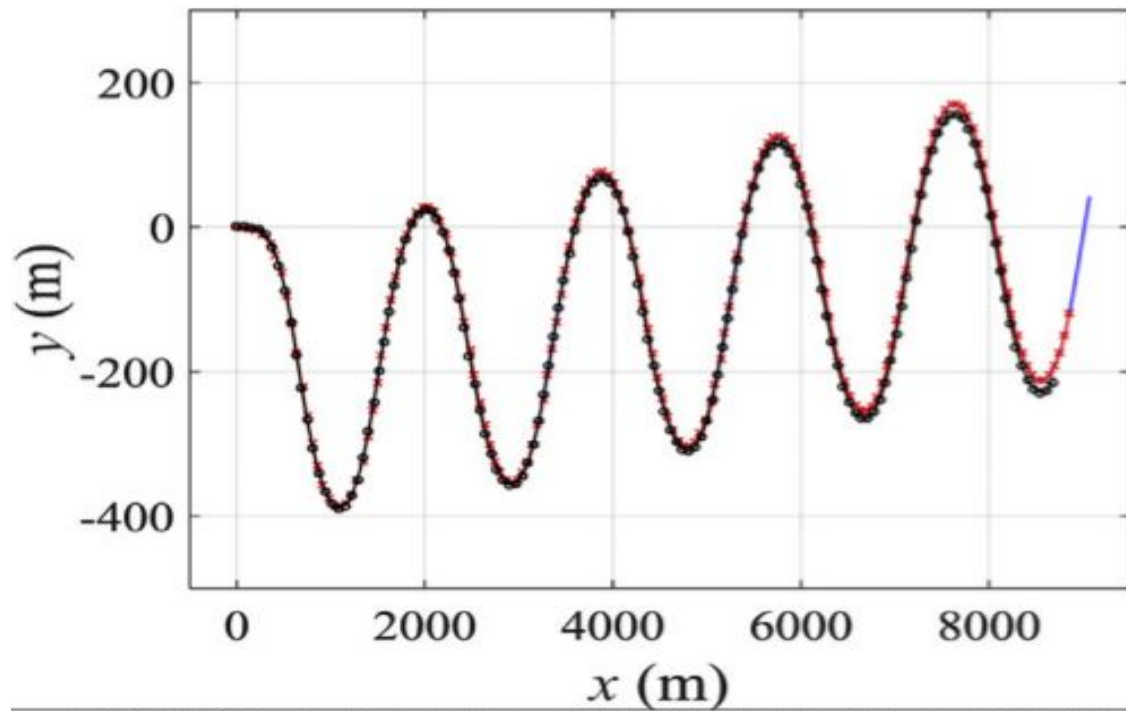
El algoritmo desarrollado es capaz de controlar la nave, incluso cuando el ángulo es 45° , mientras que el algoritmo TPI falla por completo (oscilando sobre la ruta objetivo con un error de desplazamiento creciente). Se cumple lo mismo para ángulos de 90°

C) Control sobre caminos curvos

El algoritmo desarrollado se prueba en maniobras de giro de círculo y maniobras en zigzag.



El algoritmo tiene errores de desplazamiento inicial más altos, pero reduce rápidamente los errores a valores casi cero, mientras que el algoritmo TPI tiene errores de desplazamiento que oscilan con una amplitud constante.



(a) Target path (—), present algorithm generated path(—x—), TPI algorithm generated path (—o—)

La comparación de ruta es casi exacta. El máximo error de desplazamiento en el caso del algoritmo desarrollado es aproximadamente 2.3% y 8% de la longitud del barco (≈ 3.7 m, 13 m) y en el caso del algoritmo TPI aumenta gradualmente hasta un 8% y un 10% de la longitud del barco (≈ 15 m, 17 m).

Método de crossover para algoritmos genéticos interactivos para estimar preferencias multimodales

Resumen

En este artículo de investigación se usó un algoritmo genético interactivo (iGA) que genera recomendaciones de productos. Los iGA buscan solo un punto óptimo basado en el Kansei de un usuario, pero es probable que haya varios puntos óptimos. Por ejemplo: a un usuario le puede gustar una remera azul tanto como una roja. Por tanto, el propósito de este estudio es desarrollar un nuevo método de crossover para el iGA que busque múltiples puntos óptimos para múltiples preferencias de usuarios. El método propuesto estima la ubicación de un área óptima mediante un método de clustering (agrupamiento) y luego busca los valores máximos del área con un modelo probabilístico. Para confirmar su efectividad se hicieron dos experimentos (pseudo usuario y subjetivo) que veremos más adelante.

Introducción

Hoy en día, en el comercio por internet, la recomendación de productos es muy importante. El número de productos vendidos online está incrementando. Además, para aumentar las ventas, cada sitio usa técnicas de búsqueda o de recomendación para mostrar sus productos. Las técnicas de búsqueda tienen en cuenta las entradas directas del usuario (input) y retornan productos que los usuarios esperan. En contraste, debido a que las técnicas de recomendación de productos usan registros de acción de los usuarios para analizar sus necesidades, estos muestran productos que el usuario no espera. Por lo general, las principales técnicas de recomendación son el filtrado basado en contenidos y el filtrado colaborativo. El primero recomienda productos haciendo coincidir el perfil de un usuario y los registros de acciones con las características de los productos, mientras que el segundo recomienda productos en función de la frecuencia con la que se compran al mismo tiempo. El objetivo del método propuesto es mostrar productos que se amolden a un modelo Kansei personalizado. Kansei es un término japonés relacionado con características humanas tales como la sensibilidad, la percepción, el afecto o la subjetividad. Suponemos que el Kansei humano se modela como una función. Los parámetros de entrada de la función son las características de los objetos o los factores del entorno y los parámetros de salida son evaluaciones subjetivas como preferencias o impresiones.

Este modelo Kansei humano se puede analizar con el registro de acciones del usuario. Por lo tanto, al buscar el punto máximo de esta función, podemos encontrar los objetos que coinciden con las evaluaciones subjetivas.

Idea general

La figura 1 muestra en forma general el sistema de recomendación usando iGas. Los productos son evaluados de acuerdo a las preferencias del usuario, lo que le gusta o no, o lo que quiere o no quiere. El sistema analiza el registro de evaluación de éste y selecciona

nuevos productos para mostrar, que son también evaluados por la persona. Este proceso se repite y los productos mostrados evolucionan hacia aquellos que más les gustan al usuario.

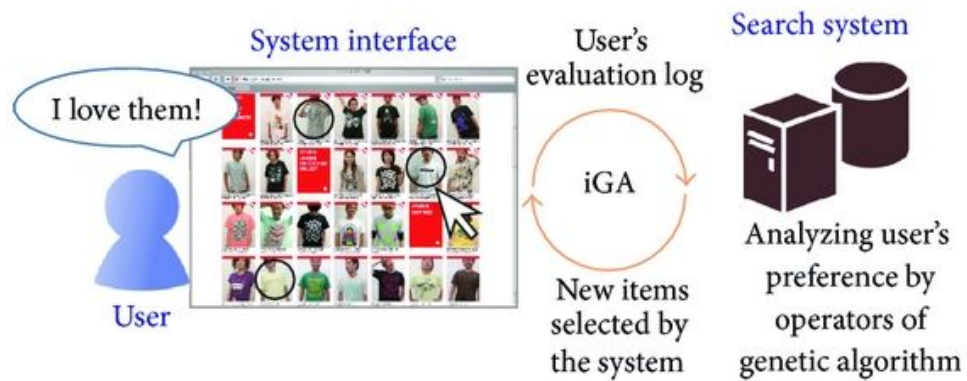
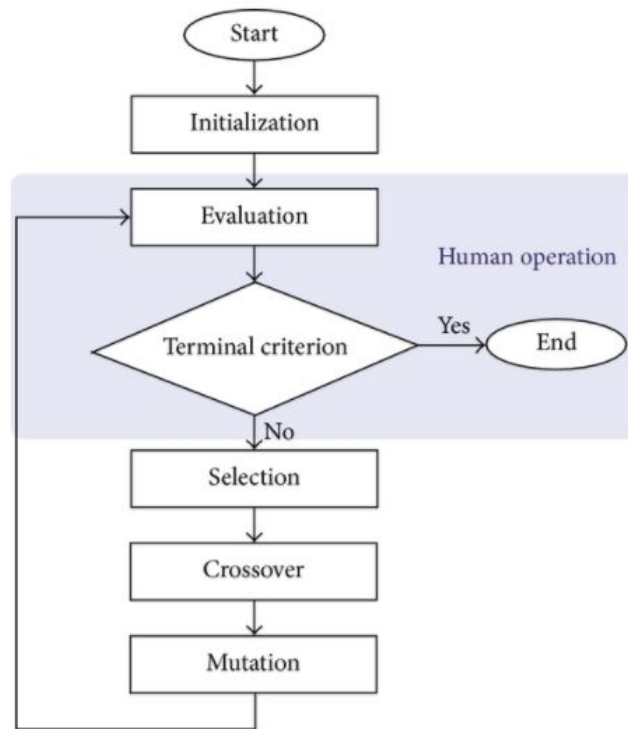


Figure 1: Overview of a product recommendation system using an iGA.

Algoritmos genéticos interactivos

Los iGas difieren de los GAs en que los valores de evaluación de las soluciones son proporcionadas por el usuario.

La figura 2 muestra un gráfico que representa el proceso de optimización de los iGAs



La representación de las soluciones difiere entre el proceso de evaluación y el resto de las operaciones. Para la evaluación (hecha por el usuario) se usa el fenotipo, que es un conjunto numérico que representa características de una solución como el color, tamaño y forma. La otra representación se llama genotipo, que es una forma codificada del fenotipo para adaptarse a los operadores del GA. En lo sucesivo, la representación de una solución se denominará variable de diseño con fines descriptivos.

Un paisaje es una función objetiva que visualiza la relación entre las variables de diseño y el valor de evaluación por rangos de montañas. Como en este estudio se usan las preferencias de un usuario como valores de evaluación y además pueden haber múltiples picos de preferencia, lo llamamos paisaje Kansei multimodal. Se puede ver una representación en las siguientes imágenes:

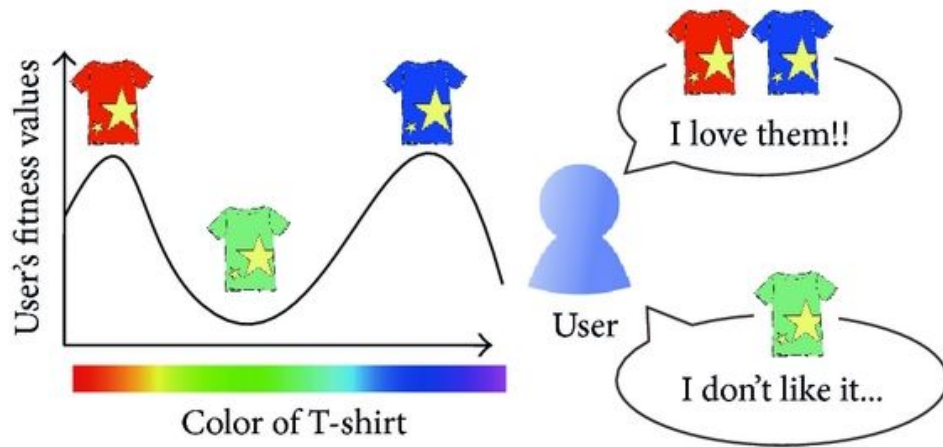
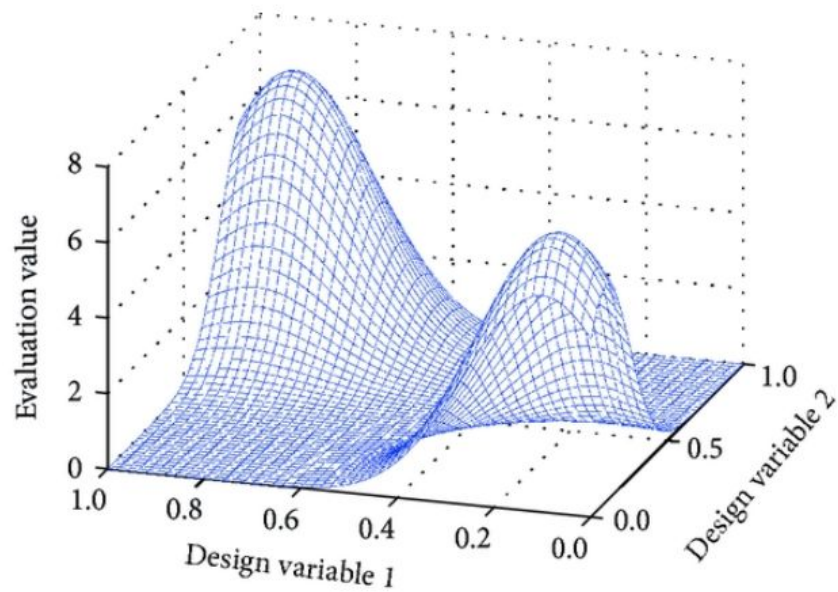


Figure 3: Landscape representation of multimodal Kansei.



Método propuesto

Se propone un nuevo método de crossover que consiste en dos etapas. Primero se agrupan soluciones con valores altos (fitness). Cada punto de datos representa una solución y cada grupo es una ubicación candidata de un pico. Aquí podemos ver un esquema:

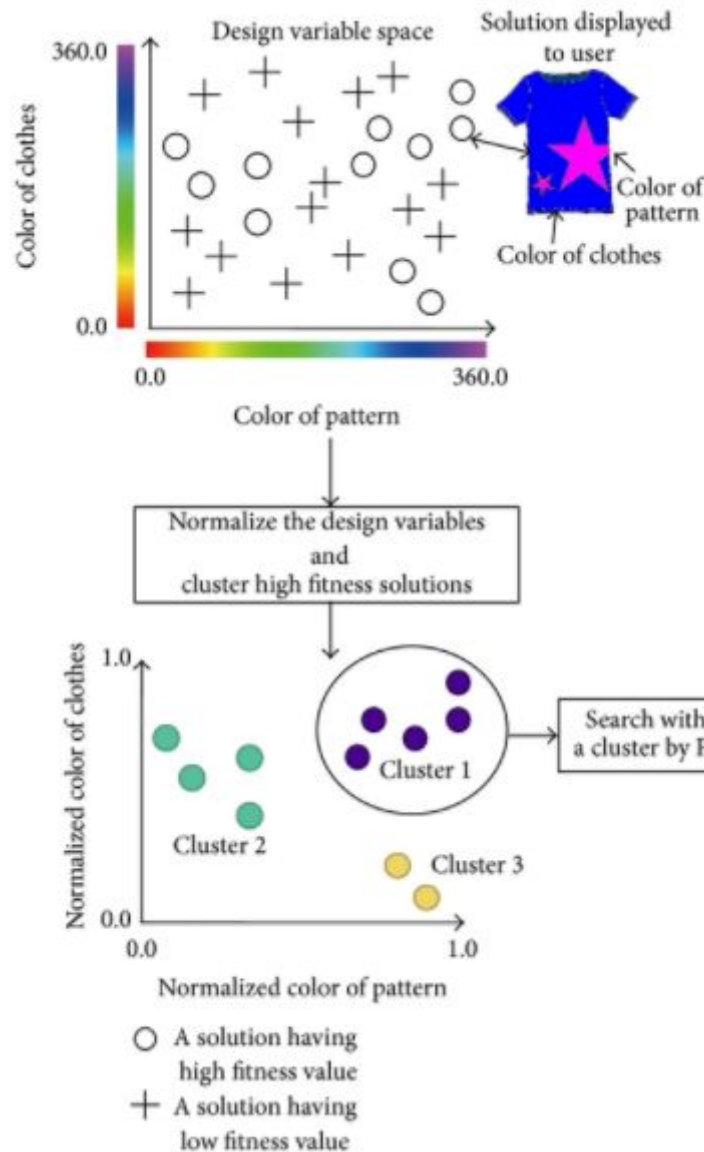


Figure 4: Example of results obtained when applying a clustering method to T-shirts solutions.

Como el número de clusters no se sabe de antemano, se usan índices que evalúan qué tan similar es un objeto a su propio clúster para saber si la configuración de agrupamiento es apropiada (silhouettes statistics).

Para buscar los valores máximos de los picos extrapolados, pasamos a la segunda etapa. En esta se generan nuevos descendientes mediante el uso de una distribución normal multidimensional construida mediante el análisis de componentes principales (técnica de la estadística para reducir la dimensión de un conjunto de datos). La Figura 5 muestra el procedimiento:

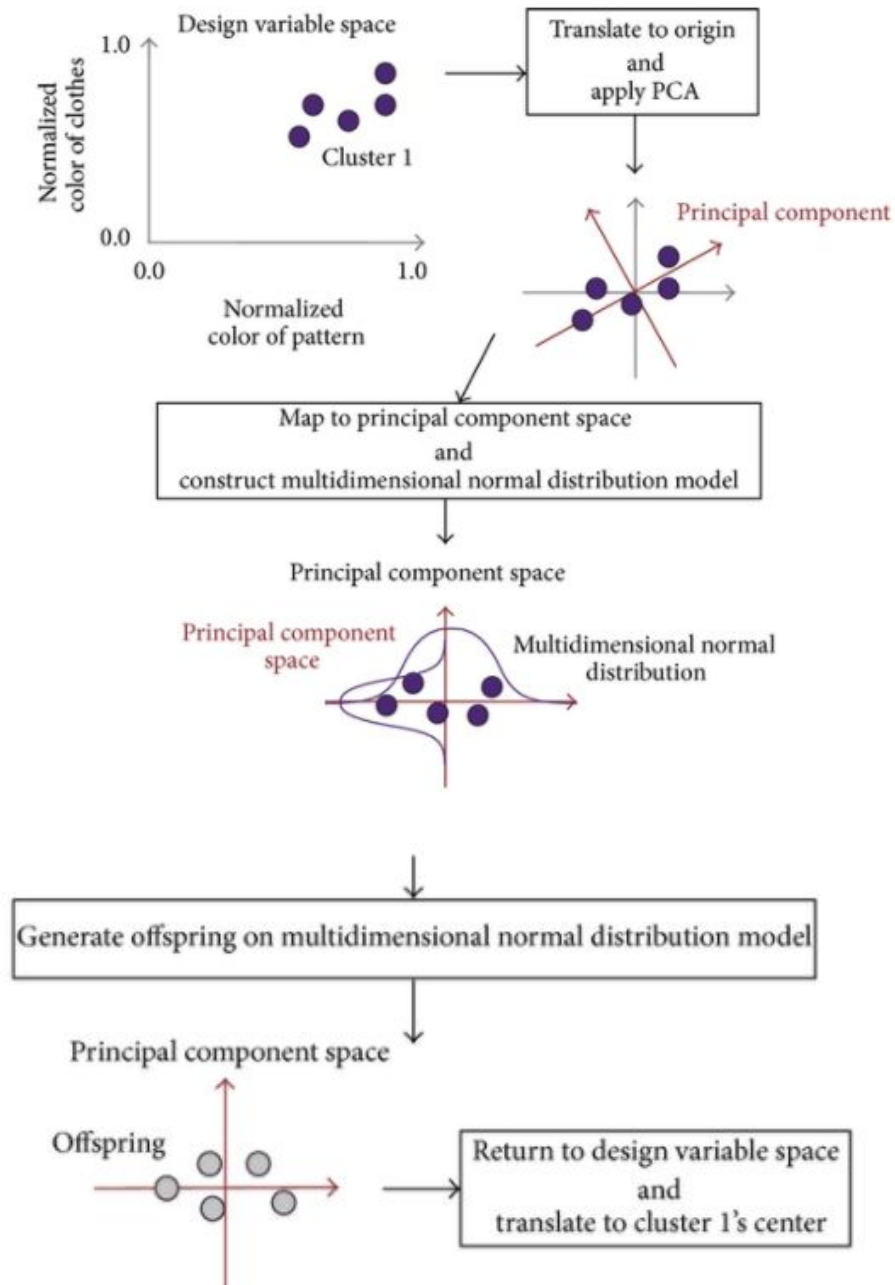


Figure 5: Procedure of offspring generation by principal component analysis.

Resultados. Experimento pseudo usuario

En este se implementó el nuevo crossover vs los métodos convencionales. El evaluador no es una persona real, sino un software para poder realizar un gran número de evaluaciones.

Se investigó el comportamiento del método propuesto variando el número de dimensiones y picos.

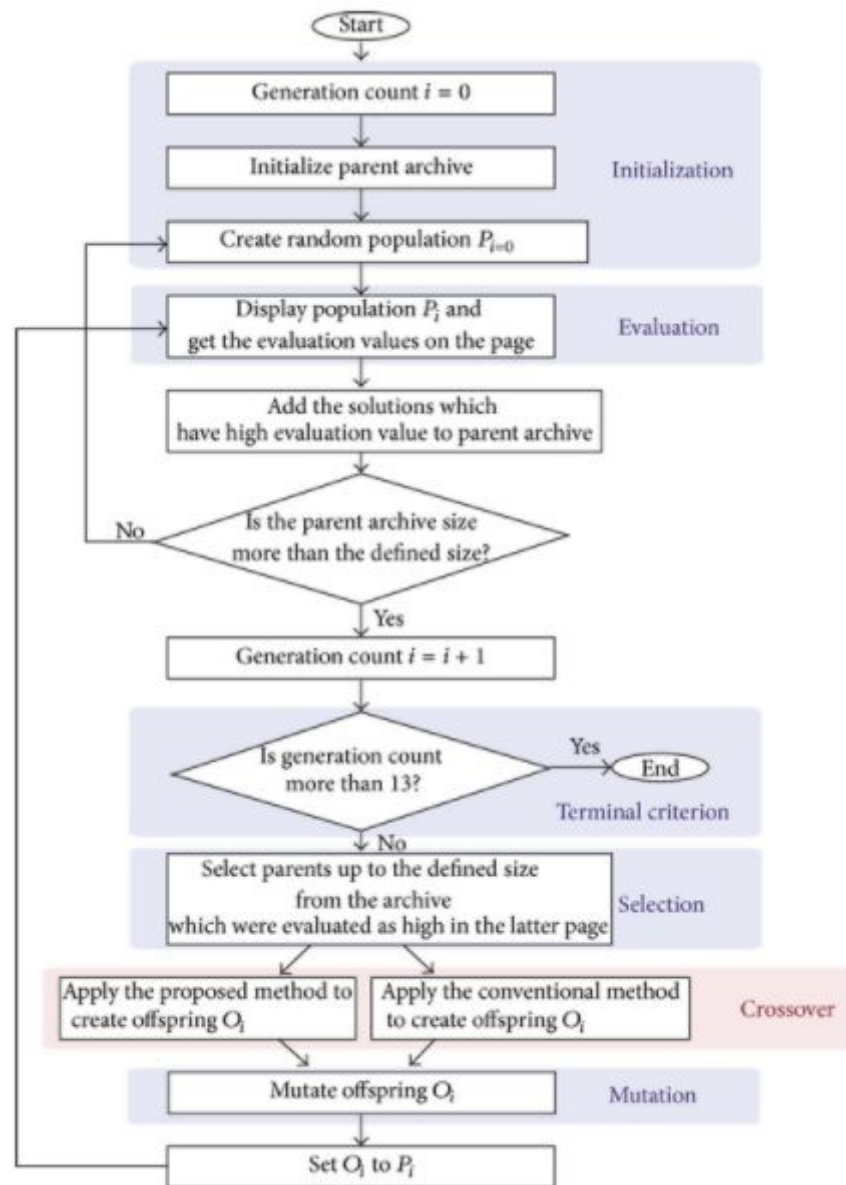


Figure 6: Flowchart of experimental system.

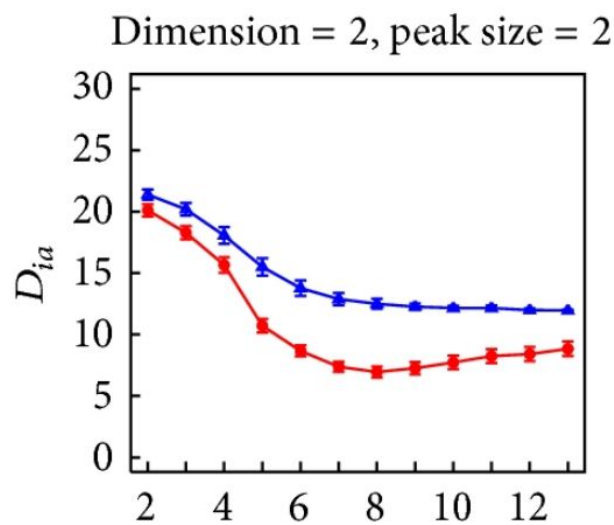
Table 1: Experimental parameters.

Parameter	Value
Population size	25
Generation size	13
Selection size	13
Crossover rate	1.0
Mutation size	5
Mutation method	Uniform mutation

En el sistema convencional se usó mezcla cruzada (blend crossover).

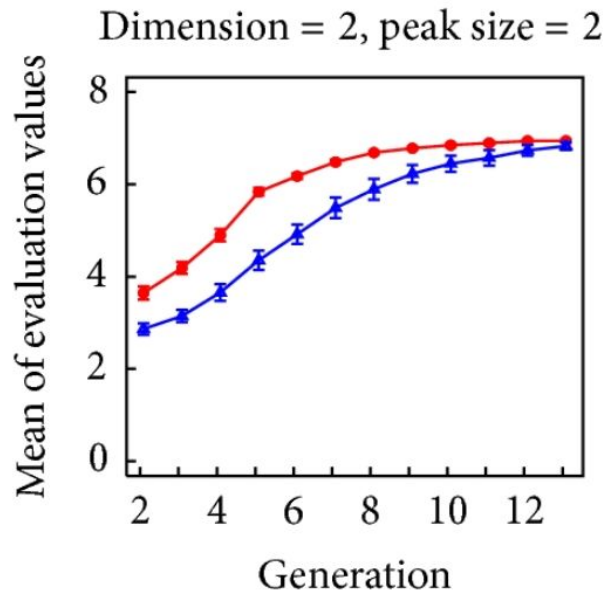
Se utilizaron dos métricas para comparar los métodos, la varianza y el perfeccionamiento.

-Varianza (D_{ia}): suma de las diferencias entre el número de descendientes generados en un pico y el número de descendientes que deberían generarse en todo el pico. Sirve para ver si el método podría identificar más picos



El método propuesto es la roja.

-Mejora: media de los valores de evaluación de una población. Los resultados de mayor calidad tienen valores más grandes.



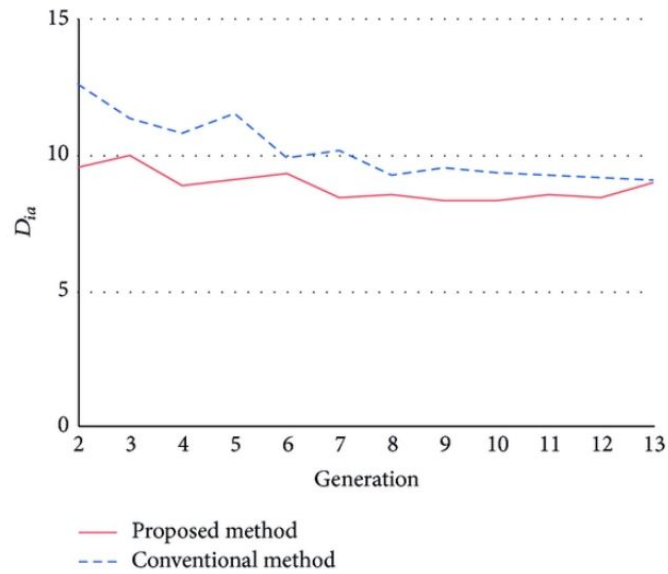
Para dimensiones más bajas, el método propuesto se desempeñó mejor que el método convencional estadísticamente en ambas métricas.

Resultados. Experimento subjetivo

En este caso el evaluador no es un pseudo-usuario sino un usuario real. Participaron ocho varones y 4 mujeres de 20 años. Se realizó un experimento preliminar para obtener los paisajes aproximados de los sujetos que participaron en el experimento. La aplicación experimental para la evaluación de preferencias es un diseño de patrón de muebles. Los resultados demuestran que el método propuesto es efectivo en humanos. El método es apropiado, especialmente en las primeras generaciones.

Examinamos la métrica de la varianza, que se utiliza para determinar si el método propuesto puede buscar picos múltiples en un paisaje de Kansei.

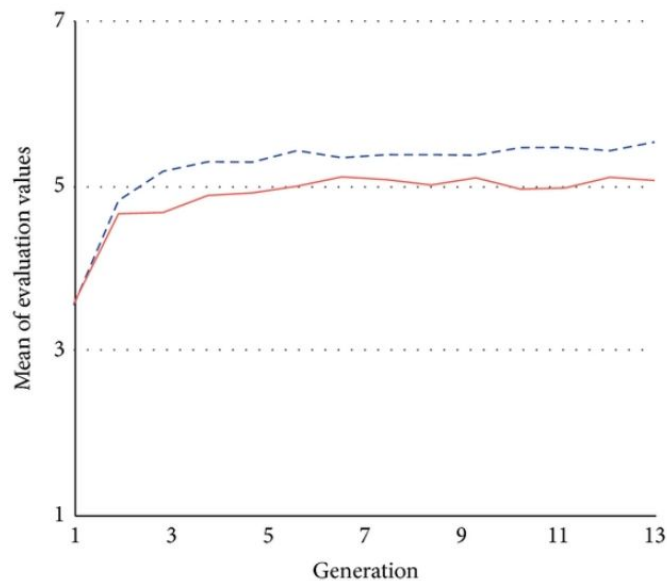
Los resultados para los paisajes con 2 y 3 picos muestran que el método propuesto obtuvo valores más pequeños, lo que indica que el método buscó más picos. Los paisajes con 2 o 3 picos son aproximadamente la mitad de los paisajes. Por lo tanto, el método propuesto es eficiente para aproximadamente la mitad de los casos considerados. Por el contrario, el método convencional fue superior para los paisajes con 4 picos.



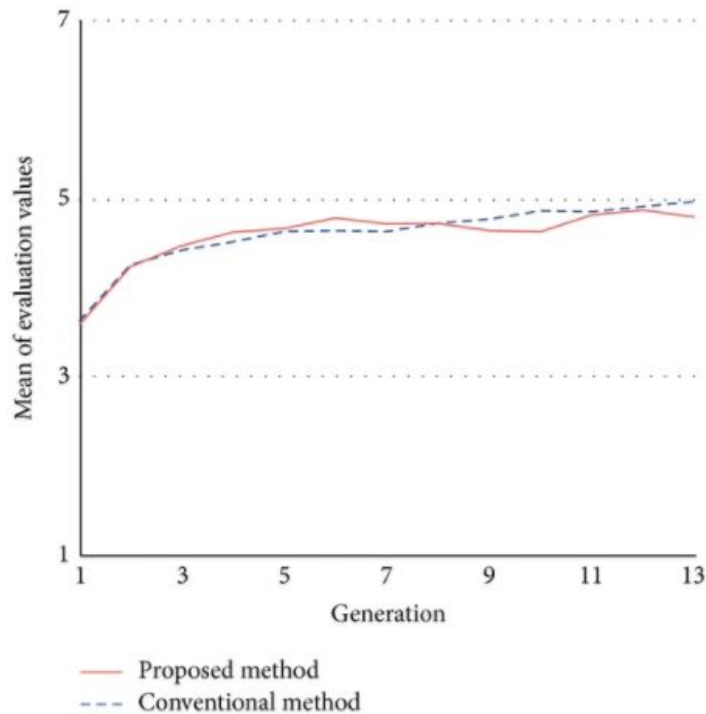
En segundo lugar, consideramos la métrica de Mejora, que se utiliza para determinar si el método propuesto buscó los valores máximos de los picos.

En la figura (a), los resultados de búsqueda de paisajes unimodales muestran que el método convencional logró valores medios más altos. Por el contrario, en la figura (b), los resultados de búsqueda de los paisajes multimodales muestran que los métodos propuestos y convencionales alcanzan aproximadamente los mismos valores de evaluación

a)



b)



<https://www.hindawi.com/journals/acisc/2013/302573/>

Programación de tareas en una celda de manufactura

Resumen

El propósito de este trabajo fue plantear una metodología para la programación de tareas en una celda de manufactura, basada en algoritmos genéticos. La celda de manufactura fue modelada como una línea de flujo y se desarrollaron cálculos que permiten saber el makespan usando una heurística adaptada de una librería realizada para algoritmos genéticos, implementado en c++.

Ambiente de producción

El problema de programación de tareas trata sobre la asignación de recursos limitados a ciertas tareas a través de determinado periodo de tiempo. Una programación adecuada puede reducir significativamente los costos de producción y reducir el tiempo de proceso permitiendo cumplir con los compromisos de entrega a tiempo. La mayoría de estos problemas son NP-hard. Un problema se considera NP-hard cuando el algoritmo aumenta su tiempo de ejecución, en el peor de los casos, de forma exponencial al tamaño del

problema. Se considera que el número de tareas y el de agentes (maquinas) es finito. La nomenclatura a utilizar es la siguiente:

n: numero de tareas

m: numero de maquinas

j: índice que se refiere a las tareas

i: índice que se refiere a las máquinas o agentes

La siguiente notación se asocia con el trabajo j:

p_{ij} : tiempo de procesamiento

r_{ij} : fecha de liberación

d_j : fecha deseada

w_j : ponderacion

Los rangos de trabajo son pequeños, complejos con un ajuste a la medida de las situaciones de un taller para altas velocidades de proceso.

Modelo del problema

Los objetivos que se persiguen en la programación de eventos en la celda de manufactura son:

- Cumplir con las fechas de vencimiento o deseadas
- Minimizar plazos
- Minimizar el tiempo o costo de preparación
- Minimizar el inventario del trabajo en proceso
- Maximizar la utilización de las máquinas o de la mano de obra
- Realizar la programación de eventos en el controlador jerárquico de la celda

El objetivo del programa es entonces, encontrar una secuencia de las tareas que minimice el máximo tiempo de finalización de todos los trabajos a ejecutar, es decir: $C_{\max} = \max_j C_j$, también llamado makespan.

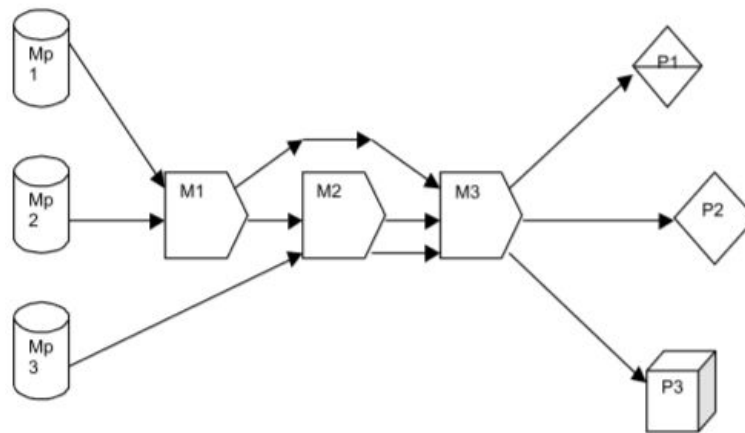


Figura 1. Secuencia en la cual se realiza un producto a lo largo

La aproximación basada en algoritmos genéticos se formalizo debido a las ventajas de estos tienen frente a otras herramientas como lógica difusa y redes neuronales, que presentan dificultades para la adecuación de las reglas de programación de tareas y la selección correcta de las mismas de acuerdo a la situación que se presente y tomar las acciones a realizar con relación a una “lista” de prioridades.