

Técnicas de optimización

Una introducción a metaheurísticas de población múltiple y algoritmos de colonia de hormigas

Juan Felipe Botero



Departamento de Ingeniería
Universidad de Antioquia
Medellín Colombia

maestriaudea.trafico@gmail.com

ÍNDICE

P-Metaheurísticas

- Motivation

- Conceptos

- Conceptos comunes de búsqueda

Optimización por colonia de hormigas

- Inteligencia de enjambre

- Colonia de hormigas: Cómo funciona?

- Elementos de solución

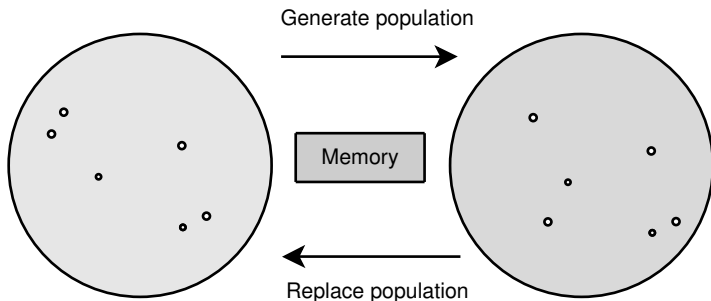
- Ejemplo: ACO para el TSP

INTRODUCCIÓN

- ▶ Las metaheurísticas de población múltiple (también llamadas **P-metahurísticas**) pueden ser vistas como una mejora iterativa de un conjunto de soluciones
- ▶ Los algoritmos que pertenecen a esta clase de metaheurísticas son:
 - ▶ Algoritmos evolutivos (AEs),
 - ▶ Búsqueda dispersa (Scatter Search),
 - ▶ Estimación de algoritmos distribuidos,
 - ▶ Optimización de enjambres de partículas (Particle Swarm Optimization PSO),
 - ▶ Colonia de abejas y de hormigas y
 - ▶ Sistemas inmunes artificiales

PRINCIPIOS FUNDAMENTALES

- ▶ Las P-Metaheurísticas básicamente realizan los siguientes pasos:
 - ▶ Su punto de partida es desde un conjunto inicial de soluciones,
 - ▶ de manera iterativa se genera una nueva población de soluciones y se
 - ▶ reemplaza la población actual con la nueva que se genera



P-METAHEURÍSTICAS: FASES

- ▶ **Fase de generación** → Una nueva población de soluciones se crea
- ▶ **Fase de reemplazo** → Una selección de las soluciones actuales se lleva a cabo a partir de la población de soluciones actuales y de las nuevas generadas
- ▶ Estas fases son llevadas a cabo mediante las iteraciones que finalizan hasta que un criterio de parada se alcanza
- ▶ La generación y el reemplazo son *sin memoria* si se basan solamente en la población actual
- ▶ De otra manera (*con memoria*), se usa la historia de la búsqueda que está guardada en la memoria

P-METAHEURÍSTICAS: PLANTILLA

High-level template of P-metaheuristics.

$P = P_0$; /* Generation of the initial population */

$t = 0$;

Repeat

 Generate(P'_t); /* Generation a new population */

$P_{t+1} = \text{Select-Population}(P_t \cup P'_t)$; /* Select new population */

$t = t + 1$;

Until Stopping criteria satisfied

Output: Best solution(s) found.

MEMORIA DE BÚSQUEDA

- **Memoria de búsqueda** → Representa el conjunto de información extraída y memorizada durante la búsqueda

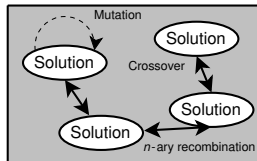
| P-metaheuristic | Search Memory |
|--|---|
| Evolutionary algorithms | Population of individuals |
| Scatter search | Population of solutions, reference set |
| Ant colonies | Pheromone matrix |
| Estimation of distribution algorithms | Probabilistic learning model |
| Particle swarm optimization | Population of particles, best global and local solutions |
| Bee colonies | Population of bees |
| Artificial immune systems: clonal selection | Population of antibodies |

FASE DE GENERACIÓN

- ▶ De acuerdo con la estrategia de generación, las P-metaheurísticas pueden ser clasificadas en dos categorías principales
 - ▶ **Basadas en evolución**
 - ▶ **Basadas en Pizarra (Blackboard based)**

P-METAHEURÍSTICAS BASADAS EN EVOLUCIÓN

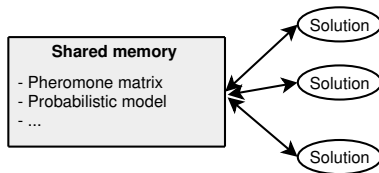
- ▶ Las soluciones que componen la población son seleccionadas y reproducidos usando los operadores de variación actuando *directamente* en sus representaciones
- ▶ Cada nueva solución se construye a partir de los diferentes atributos pertenecientes a la población actual



Evolutionary-based P-metaheuristics:
evolutionary algorithms, scatter search, ...

P-METAHEURÍSTICAS BASADAS EN PIZARRA

- ▶ Aquí, las soluciones de la población participan de manera conjunta en la construcción de la memoria compartida
- ▶ Esta memoria compartida será la entrada principal para generar la nueva población de soluciones



Blackboard-based P-metaheuristics:
ant colonies, estimation distribution algorithms, ...

FASE DE SELECCIÓN

- ▶ El último paso de las P-metaheurísticas consiste en seleccionar las nuevas soluciones a partir de la unión de la población de soluciones actuales y las generadas
- ▶ Estrategia tradicional → Se selecciona la población generada como la nueva generación
- ▶ Estrategia alternativa → Usar algún tipo de *elitismo* (mejores soluciones de los dos conjuntos)
- ▶ En las P-metaheurísticas basada en pizarra, **No hay** un método explícito de selección. La nueva población de soluciones podrá actualizar la memoria compartida que afectará la generación de la población subsiguiente

POBLACIÓN INICIAL

- ▶ La determinación de la población inicial es muy a menudo mirada con indiferencia en el diseño de una P-metaheurística
- ▶ Sin embargo, este paso juega un rol crucial en la efectividad del algoritmo y en su eficiencia
- ▶ El criterio principal que se debe tener en cuenta en la generación de la población inicial, es la diversificación:
 - ▶ Si la población inicial no está bien diversificada, una convergencia prematura puede ocurrir para cualquier P-Metaheurística

POBLACIÓN INICIAL: GENERACIÓN ALEATORIA

- ▶ Usualmente, la población inicial es generada de manera aleatoria
- ▶ Por ejemplo, en la optimización continua, el valor inicial de cada variable de la solución puede ser generada de manera aleatoria en su rango posible (que es definido por las restricciones del problema)
- ▶ La forma más popular de generación aleatoria de la población es la *pseudo-aleatoria*
- ▶ También existe la generación *quasi-aleatoria* donde el objetivo del generador no solamente se relaciona con la independencia entre las soluciones sucesivas pero también con su dispersión

POBLACIÓN INICIAL: DIVERSIÓN SECUENCIAL

- ▶ Algunos de los procedimientos de inicialización no requieren ninguna evaluación de la función objetivo o de las restricciones asociadas al problema de optimización
- ▶ En la diversificación secuencial, las soluciones son generadas en secuencia de tal manera que la diversidad es optimizada

POBLACIÓN INICIAL: DIVERSIFICACIÓN PARALELA

- En la estrategia de diversificación paralela, las soluciones de una población son generadas de manera independiente y paralela

INICIALIZACIÓN HEURÍSTICA

- ▶ Cualquier heurística (e.g., búsqueda local) puede ser usado para inicializar la población de soluciones
- ▶ Incluso una heurística de tipo voraz (greedy) puede ser usada para determinar la solución inicial
- ▶ La principal dificultad de este tipo de inicialización es que la población inicial puede perder su diversidad, y generará una convergencia prematura y un estancamiento de la población

TÉCNICAS DE INICIALIZACIÓN: UNA COMPARACIÓN

| Strategy | Diversity | Computational Cost | Quality of Initial Solutions |
|----------------------------|-----------|--------------------|------------------------------|
| Pseudo-random | ++ | +++ | + |
| Quasi-random | +++ | +++ | + |
| Sequential diversification | ++++ | ++ | + |
| Parallel diversification | ++++ | +++ | + |
| Heuristic | + | + | +++ |

CRITERIO DE PARADA

- ▶ Muchos criterios de parada basados en la evolución de la población pueden ser usados. Algunos de ellos son similar a aquellos designados por las S-metaheurísticas:
 - ▶ **Procedimiento estático** → El fin de la búsqueda puede ser conocida *a priori* (e.g. número fijo de iteraciones)
 - ▶ **Procedimiento adaptativo** → En un procedimiento adaptativo, el fin de la búsqueda no puede ser conocido *a priori*
 - ▶ Otro criterio es el parar la P-metaheurística cuando la medida de diversidad cae bajo un umbral

INTELIGENCIA DE ENJAMBRE

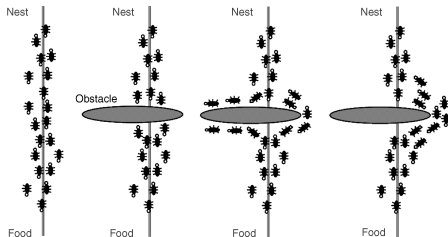
- ▶ Los algoritmos inspirados por el comportamiento colectivo de las especies tales como: hormigas, abejas, avispa, termitas, peces y aves se denominan como algoritmos de inteligencia de enjambre
- ▶ La inteligencia de enjambre se origina del comportamiento social de esas especies que compiten por comida
- ▶ Las principales características de los algoritmos basados en la inteligencia de enjambre es su división en partículas que actúan como agentes simples y no sofisticados. Estas partículas cooperan mediante de un medio indirecto de comunicación, y realizan movimientos en el espacio de decisión
- ▶ Entre los métodos más exitosos de los algoritmos de optimización basados en la inteligencia de enjambre son **colonia de hormigas** y **optimización de enjambre de partículas** (particle swarm optimization)

OPTIMIZACIÓN POR COLONIA DE HORMIGAS (ACO): INTRODUCCIÓN

- ▶ Idea básica → Imitar el comportamiento cooperativo de las hormigas para resolver problemas de optimización
- ▶ Las hormigas pueden ser vistas como un sistema multiagentes en donde cada agente simple está inspirado por el comportamiento de una hormiga
- ▶ Algoritmos ACO → Una colonia de hormigas es capaz de encontrar el camino más corto entre dos puntos usando un mecanismo de comunicación muy simple

ACO: CÓMO FUNCIONA? (I)

- ▶ Durante los viajes de las hormigas, un rastro químico (feromona) se deja en tierra
- ▶ El rol del rastro es guiar las otras hormigas hacia el punto objetivo
- ▶ Mientras más grande sea la cantidad de feromona de un camino particular, mayor es la probabilidad de que las hormigas seleccionen ese camino
- ▶ Una hormiga dada escoge el camino de acuerdo a la cantidad de feromona que huele



ACO: CÓMO FUNCIONA? (II)

- ▶ La feromona tiene una acción decreciente a través del tiempo
- ▶ La cantidad de feromona dejada por una hormiga depende de la cantidad de comida que consigue (proceso de refuerzo)
- ▶ El comportamiento colaborativo de las hormigas se conoce como *estigmergia*

ACO: PLANTILLA DE ALGORITMO

Template of the ACO.

Initialize the pheromone trails ;

Repeat

For each ant **Do**

 Solution construction using the pheromone trail ;

Update the pheromone trails:

 Evaporation ;

 Reinforcement ;

Until Stopping criteria

Output: Best solution found or a set of solutions.

ACO: RESUMEN

- ▶ Cuándo se diseña un algoritmo ACO los elementos más importantes a determinar son:
 - ▶ **Información de feromona**→ Consiste en la definición de un vector de parámetros de modelo τ llamado paramentros del *rastro de feromona*. Los valores de feromona $\tau_i \in \tau$ deben reflejar la información relevante en la construcción de la solución para un problema dado. Estos valores están usualmente asociados con los componentes de una solución
 - ▶ **Construcción de la solución**→ La cuestión principal aquí tiene que ver con la definición de la heurística local a ser usada en guiar la búsqueda, en adición a la feromona
 - ▶ **Actualización de feromona**→ La parte más importante es decidir la estrategia de aprendizaje por refuerzo

CONSTRUCCIÓN DE SOLUCIÓN (I)

- ▶ La construcción de soluciones se hace de acuerdo a una regla de transición basada en un estado probabilístico
- ▶ Las hormigas pueden ser consideradas como procedimientos voraces y estocásticos
- ▶ Las hormigas construyen una solución de manera probabilística añadiendo componentes de la solución a unos componentes parciales hasta que una solución completa se logre

CONSTRUCCIÓN DE SOLUCIÓN (II)

- ▶ El objetivo de optimización del problema se puede ver como un grafo de decisión dónde una hormiga construye un camino
- ▶ El proceso es iterativo y considera:
 - ▶ **Rastros de feromona** → La feromona memoriza las características de las “buenas” soluciones generadas, éstas guiarán la construcción de nuevas soluciones por las hormigas. Los rastros de feromona cambian dinámicamente durante la búsqueda para reflejar el conocimiento adquirido
 - ▶ **Información dependiente del problema** → La información específica del problema ayuda a las hormigas en sus decisiones para construir mejores soluciones

ACTUALIZACIÓN DE FEROMONA

- Para actualizar la feromona, se usan las soluciones generadas. Una regla global de actualización de feromona es aplicada en dos fases:

- *Fase de evaporación* → El rastro de la feromona se decrementa automáticamente. Cada valor de la feromona se reduce en una proporción fija

$$\tau_i = (1 - \rho)\tau_i, \forall i \in [1, n]$$

dónde $\rho \in (0, 1]$ representa la tasa de reducción de la feromona. La evaporación previene una convergencia prematura para todas las hormigas a soluciones que son “buenas” (óptimos locales) y fomenta la diversificación en el espacio de búsqueda

- *Fase de refuerzo* → Actualiza el rastro de la feromona de acuerdo a las soluciones generadas

ACTUALIZACIÓN DE FEROMONA: FASE DE REFUERZO (I)

- ▶ Existen tres alternativas diferentes que pueden ser aplicadas en la fase de refuerzo
 - ▶ *Actualización de feromona paso a paso* → El rastro de la feromona τ_i se actualiza por una hormiga a cada paso de la construcción de la solución
 - ▶ *Actualización de feromona retrasada* → La actualización de la feromona τ se aplica cuando la hormiga genera la solución completa

ACTUALIZACIÓN DE FEROMONA: FASE DE REFUERZO (II)

- ▶ *Actualización de feromona fuera de linea*→ Esta es la alternativa más popular y actualiza el rastro de la feromona sólomente cuándo **cada una** de las hormigas genera una solución completa. Diferentes estrategias se pueden usar de esta manera:
 - ▶ **Actualización de feromona basada en calidad**→ Esta estrategia actualiza el valor de la feromona asociado con la mejor solución encontrada entre todas las hormigas
 - ▶ **Actualización basa en ranking**→ Las mejores k soluciones pueden actualizar la feromona en una cantidad que depende del ranking que hayan tenido
 - ▶ **Peor actualización de feromona**→ La hormiga que genera la peor solución decrementará los rastros de feromona que están asociados con los componentes de esa solución
 - ▶ **Actualización de la feromona elitista**→ La mejor solución que se ha encontrado en todo la búsqueda actualizará la feromona para promover una intesificación de la búsqueda

EJEMPLO: ACO PARA EL TSP

- ▶ Si $G = (V, E)$ es el grafo de entrada
- ▶ El diseño de un algoritmo ACO para el TSP necesita →
Definición de los rastros de la feromona y el procedimiento de construcción de la solución

ACO PARA EL TSP: RASTROS DE LA FEROMONA

- ▶ La feromona τ_{ij} estará asociada con el enlace (i, j) del grafo G
- ▶ La información de la feromona puede ser presentada por la matriz τ de $n \times n$ dónde cada elemento de la matriz (τ_{ij}) expresa la deseabilidad de tener el enlace (i, j) en el tour
- ▶ Durante la búsqueda, la feromona es actualizada para estimar la utilidad de cada enlace del grafo

ACO PARA EL TSP: CONSTRUCCIÓN DE LA SOLUCIÓN

- Cada hormiga construye un tour de manera estocástica. Dada una ciudad inicial arbitraria i , una hormiga seleccionará la próxima ciudad j con la siguiente probabilidad

$$p_{ij} = \frac{\tau_{ij}}{\sum_{k \in S} \tau_{ik}}, \forall j \in S$$

dónde S representa el conjunto de ciudades que aún no se han visitado de G . Las hormigas pueden usar una ciudad inicial escogida aleatoriamente en la fase de construcción

- Conocimiento adicional específico del problema → Consideremos los valores ν_{ij} iguales a $1/d_{ij}$ dónde d_{ij} es la distancia entre las ciudades i y j . El computar la decisión de las probabilidades de transición, p_{ij} se actualiza como sigue:

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \nu_{ij}^{\beta}}{\sum_{k \in S} \tau_{ik}^{\alpha} \cdot \nu_{ik}^{\beta}}, \forall j \in S$$

dónde α y β son parámetros que representan la influencia de los valores de la feromona y de los valores dependientes del problema

- $\alpha = 0$ → La información de la feromona no se tiene en cuenta y las ciudades más cercanas son las que más probablemente son seleccionadas,
- $\beta = 0$ → Sólo los rastros de la feromona guían la búsqueda (en este caso puede suceder que todas las hormigas construyan el mismo tour subóptimo). Por lo tanto, un buen compromiso debe ser hayado entre

α y β

ACO PARA EL TSP: ACTUALIZACIÓN DE FEROMONA

- Actualización de feromona → Cada hormiga incrementará las feromonas de manera proporcional a la calidad del tour π obtenido

$$\tau_{i\pi(i)} = \tau_{i\pi(i)} + \Delta, \forall i \in [1, n]$$

dónde $\Delta = 1/f(\pi) \rightarrow$ buenos tours emergen como el resultado de la cooperación entre hormigas a través de los rastros de feromona

- Procedimiento de evaporación → Para cada enlace del grafo, la feromona $\tau_{i,j}$ se evaporará como sigue:

$$\tau_{ij} = (1 - \rho)\tau_{ij}, \forall i, j \in [1, n]$$

dónde $\rho \in (0, 1]$ es la tasa de reducción de la feromona

ACO PARA EL TSP: PLANTILLA DEL ALGORITMO

Ant colony algorithm for the TSP problem (ACO–TSP).

Initialize the pheromone information ;

Repeat

For each ant **Do**

 Solution construction using the pheromone trails:

$S = \{1, 2, \dots, n\}$ /* Set of potentially selected cities */

 Random selection of the initial city i ;

Repeat

 Select new city j with probability $p_{ij} = \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{k \in S} \tau_{ik}^{\alpha} \times \eta_{ik}^{\beta}}$;

$S = S - \{j\}$; $i = j$;

Until $S = \emptyset$

End For

Update the pheromone trail:

For $i, j \in [1, n]$ **Do**

$\tau_{ij} = (1 - \rho)\tau_{ij}$ /* Evaporation */ ;

For $i \in [1, n]$ **Do**

$\tau_{i\pi(i)} = \tau_{i\pi(i)} + /$ * π : best found solution */ ;

Until Stopping criteria

Output: Best solution found or a set of solutions.

BIBLIOGRAPHY

- E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley & Sons, Hoboken, New Jersey, 2009.