

Multi-Objective taxi ride sharing

Diego D. Charrez Ticona¹

¹Departamento de Ciencia de la Computación
Universidad Nacional de San Agustín

Computación Bioinspirada, 2018

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Problema

El objetivo es minimizar el costo total de todos los pasajeros y el retraso percibido de los pasajeros para llegar a su destino.

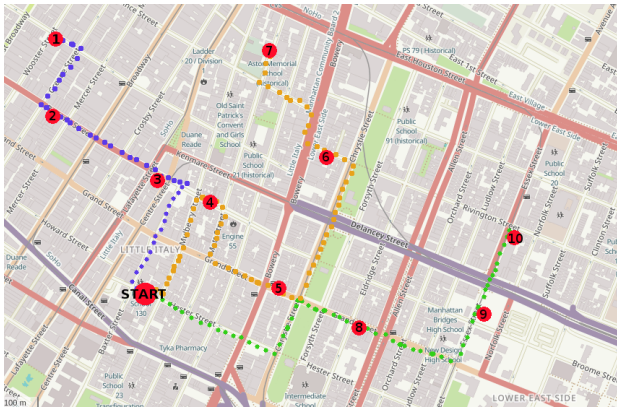


Figura: Rutas de taxis de viaje compartido

Problema

El problema modela siguiente situación:

- ▶ N personas están en el mismo origen.
- ▶ Se movilizaran a diferentes destinos compartiendo taxis.
- ▶ La asignación del taxi debe ser hecha de tal forma que se minimiza el costo total.

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Restricciones

- ▶ Cada taxi tiene un numero limitado de pasajeros.
- ▶ El máximo numero de taxis para N pasajeros es N .

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Valores de entrada

- ▶ Numero de pasajeros.
- ▶ An array with indicating if a passenger is able to wait.
- ▶ An array of taxis with their capacities.
- ▶ Initial cost.
- ▶ Matriz de costos entre todos los puntos.
- ▶ Matriz de tiempo de viaje entre todos los puntos.

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

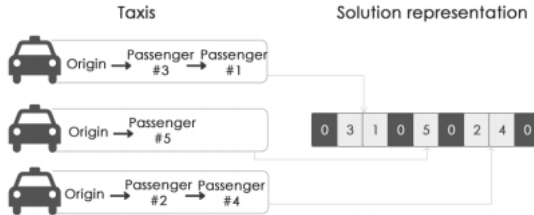
Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Codificación

Los pasajeros que aparecen entre ceros se encuentran en un mismo taxi.



Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Representación de solución del problema de viaje compartido

Codificación

Los pasajeros que aparecen entre ceros se encuentran en un mismo taxi.



Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Representación de solución del problema de viaje compartido

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Iniciación de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Algoritmo 5 Inicialización ávida de la población.

```
desde  $m = 1$  a  $2N - 1$  hacer
    población[1][m] = solución_ávida[m]; {primer individuo (copia directa)}
fin desde
desde  $i = 2$  a tamaño(población) hacer
    individuo = población[i];
    desde  $j = 1$  a  $2N - 1$  hacer
        individuo[j] = solución_ávida[j];
    fin desde
    cantidad_perturbaciones = aleatorio(1,  $\frac{2N-1}{4}$ );
    desde  $k = 1$  a cantidad_perturbaciones hacer
        posición1 = random(1,  $2N - 1$ );
        posición2 = random(1,  $2N - 1$ );
        mientras posición2 == posición1 hacer
            posición2 = random(1,  $2N - 1$ );
        fin mientras
        individuo.intercambiar(posición1, posición2);
    fin desde
fin desde
```

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Inicialización de población

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Algoritmo 6 Procedimiento de corrección de soluciones.

```

mientras no terminar hacer
    largo_secuencia = 0;
    posición = 0;
    mientras largo_secuencia  $\leq C_{MAX}$  y posición  $< 2N - 1$  hacer
        si individuo[posición]  $\neq 0$  entonces
            largo_secuencia++;
        en otro caso
            largo_secuencia = 0;
        fin si
        posición++;
    fin mientras
    si largo_secuencia  $> C_{MAX}$  entonces
        punto_de_corte = random(posición -  $C_{MAX} + 1$ , posición - 1);
        cero = encontrarPrimerCeroConsecutivo();
        individuo.intercambiar(cero, punto_de_corte);
    en otro caso
        terminar
    fin si
fin mientras
    
```

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Corrección de soluciones

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Padre 1

2	0	1	5	4	0	0	3	0
---	---	---	---	---	---	---	---	---

Padre 2

0	3	1	0	2	0	4	0	5
---	---	---	---	---	---	---	---	---

PBX

1) Se sortean n posiciones en el padre 1

2		1	5	4				0
---	--	---	---	---	--	--	--	---

2) Se copian en el hijo los valores sorteados en el padre 1

2		1	5	4				0
---	--	---	---	---	--	--	--	---

3) Se seleccionan en el padre 2 los valores no sorteados del padre 1

0	3	1	0	2	0	4	0	5
---	---	---	---	---	---	---	---	---

4) Se copian en orden los valores seleccionados del padre 2 en las posiciones vacías del hijo

2	3	1	5	4	0	0	0	0
---	---	---	---	---	---	---	---	---

Función Correctiva

2	3	0	5	4	1	0	0	0
---	---	---	---	---	---	---	---	---

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

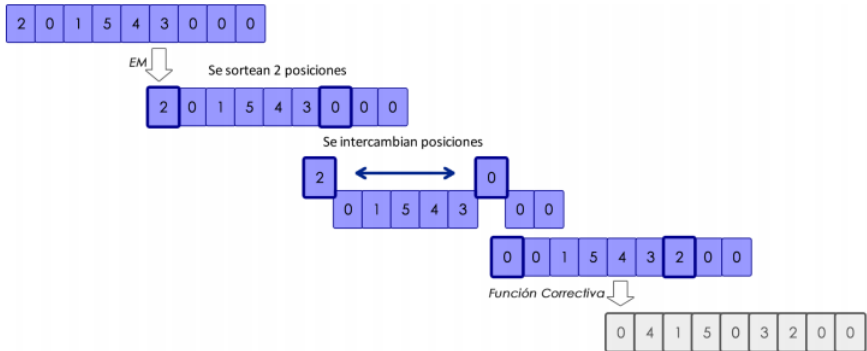
Figura: Cruzamiento PBX

Algoritmo 7 Cruzamiento basado en la posición (*PBX*).

- 1: seleccionar aleatoriamente varias posiciones del padre 1.
 - 2: generar parcialmente el hijo 1, copiando los valores (*alelos*) de las posiciones elegidas del padre 1 directamente al hijo 1.
 - 3: marcar los alelos del padre 2 que ya fueron seleccionados en el padre 1.
 - 4: desde el inicio del padre 2, seleccionar secuencialmente el siguiente alelo que no haya sido marcado, y copiarlo a la primer posición libre del hijo 1, desde el comienzo.
-

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Cruzamiento PBX



Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Mutacion por intercambio

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

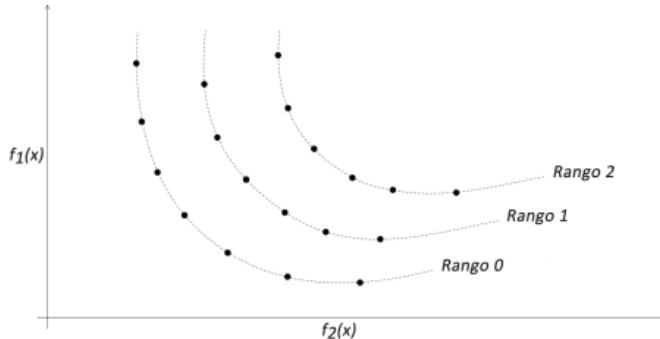
Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO



Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Dominancia y rankings

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

Algoritmo 3 Esquema del algoritmo *NSGA-II*.

```
1: inicializar( $P(0)$ )
2: generación = 0
3: Evaluar ( $P(0)$ )
4: mientras no CriterioParada hacer
5:    $R = \text{Padres} \cup \text{Hijos}$ 
6:   Frentes = Ordenamiento no Dominado( $R$ )
7:   NuevaPop =  $\emptyset$ 
8:    $i = 1$ 
9:   mientras  $|\text{NuevaPop}| + |\text{Frentes}(i)| \leq \text{sizepop}$  hacer
10:    Calcular Distancia de Crowding ( $\text{Frentes}(i)$ )
11:    NuevaPop = NuevaPop  $\cup$   $\text{Frentes}(i)$ 
12:     $i++$ 
13:  fin mientras
14:  Ordenamiento por Distancia( $\text{Frentes}(i)$ )
15:  NuevaPop = NuevaPop  $\cup$   $\text{Frentes}(i)[1:(\text{sizepop} - |\text{NuevaPop}|)]$ 
16:  Hijos = Selección y Reproducción(NuevaPop)
17:  generación++
18:   $P(\text{generación}) = \text{NuevaPop}$ 
19: fin mientras
20: retornar mejor solución
```

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Algoritmo NSGAI

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

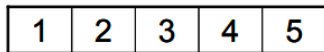
Fast non Dominated Sort

Algoritmo NSGAI

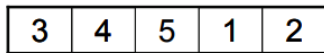
Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

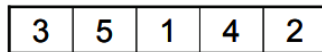
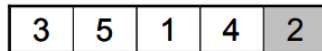
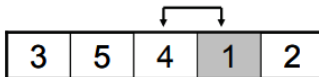
MOPSO



Origin



Rotation



Target

Source: Multiobjective taxi sharing optimization using the NSGA-II evolutionary algorithm

Figura: Substraccion entre soluciones

Outline

Problema

Problema

Restricciones

Implementación

Valores de entrada

Codificación

Inicialización de población

Corrección de soluciones

Non-Dominated Sorting Genetic Algorithm-II (NSGAI)

Cruzamiento PBX

Fast non Dominated Sort

Algoritmo NSGAI

Multi-Objective Particle Swarm Optimization Algorithm (MOPSO)

Diferencia entre soluciones

MOPSO

```
input :  $\omega, C_1, C_2, N$ 
for  $i = 1$  to  $N$  do
    Inicializar la  $i$ -ésima partícula en una velocidad y
    posición aleatoria
    evaluar( $i$ )
    Actualizar el repositorio global ( $G$ )
    Actualizar el repositorio personal de la  $i$ -ésima
    partícula ( $P_i$ )
while not condicion_de_parada() do
    for  $i = 1$  to  $N$  do
        Seleccionar mejor posición personal (elemento
        aleatorio en  $P_i$ )
        Seleccionar mejor posición global (elemento
        aleatorio en  $G$ )
        Calcular velocidad de la  $i$ -ésima partícula
        Calcular la nueva posición de la  $i$ -ésima
        partícula
    for  $i = 1$  to  $N$  do
        evaluar( $i$ )
        Actualizar el repositorio global ( $G$ )
        Actualizar el repositorio personal de la  $i$ -ésima
        partícula ( $P_i$ )
Retornar soluciones_no_dominadas( $G$ )
```

For Further Reading I



Massobrio, Renzo

International Journal of Metaheuristics, 5(1):67–90, 2016.