

A Fully Heuristic Framework for the Sociometric Multiple Team Formation Problem

Ignacio Martínez Hernández

Doctorado en Sistemas de Ingeniería
Universidad de Talca

Metahurísticas

<https://github.com/DarkNacho/MTFP>

Introducción y Motivación

- **Contexto:** Formación de equipos para múltiples proyectos concurrentes en organizaciones.
- **Objetivo:** Optimizar la asignación de recursos humanos, satisfaciendo:
 1. Requerimientos de habilidades (r_{kl}).
 2. Capacidad laboral individual ($\leq 100\%$).
 3. Afinidad sociométrica (Maximizar E).

Limitación del Estado del Arte

El enfoque original para el MTFP (Gutiérrez et al., 2016) se basó en una búsqueda híbrida que dependía de **solucionadores exactos de Programación por Restricciones (CP)** para validar la factibilidad.

Contribución Principal

Método totalmente heurístico que reemplaza el uso de solver.

El objetivo es maximizar la eficiencia global (E), una suma ponderada de las eficiencias de proyecto (e_l).

Función Objetivo: Eficiencia Sociométrica

$$E = \sum_{l \in \mathcal{P}} w_l \cdot e_l$$

$$e_l = \frac{1}{2} \left(1 + \frac{\sum_{i,j} s_{ij} \cdot x_{il} x_{jl}}{\left(\sum_k r_{kl} \right)^2} \right)$$

- s_{ij} : Matriz de afinidad sociométrica (ej. $\in \{-1, 0, 1\}$).
- $x_{il}x_{jl}$: El impacto sociométrico se escala por la dedicación simultánea.

Restricciones

- 1. Capacidad:** $\sum_l x_{il} \leq 1$
- 2. Habilidades:** $\sum_{i \in Q_k} x_{il} = r_{kl}$
- 3. Dedicación Discreta:** $x_{il} \in \mathcal{D}$ (ej. $\{0.0, 0.25, 0.5, 0.75, 1.0\}$)

Metodología: El Núcleo Heurístico

Randomized Greedy Constructive Heuristic (RGCH)

Función: Generar sub-soluciones factibles, sustituyendo la llamada al solver exacto.

- **Estrategia:** Explotación de la **Descomposición del Problema por Habilidad** (\mathcal{K}).
- **Procedimiento:**
 1. Reiniciar asignaciones para la habilidad k .
 2. Aleatorizar el orden de los candidatos de habilidad Q_k (Estocasticidad).
 3. Asignar dedicaciones discretas de forma **Greedy** hasta satisfacer el requerimiento r_{kl} sin violar la capacidad individual.

Estructura de Vecindario Fundamental (N^1)

- **Generación de Vecino (X'):**
 1. Seleccionar aleatoriamente una habilidad k .
 2. Aplicar el **RGCH** para reconstruir la asignación de esa única habilidad.
- **Propiedad Clave:** Todos los vecinos $X' \in N^1(X)$ son inherentemente factibles.

Algoritmos de Búsqueda Comparados

Se evaluaron siete algoritmos bajo un presupuesto de **Evaluaciones de Función (NFE)** escalado para garantizar una comparación justa.

Componente Unificador: N^1 y RGCH

Todos los algoritmos de búsqueda (HC, TS, SLS, VNS, GA) utilizan el **RGCH** como motor para generar:

- Movimientos de exploración (Vecinos en N^1).
- Mutaciones (en el GA).

Descripción del Algoritmo Genético (GA)

El GA se diseñó para explotar la descomposición por habilidades:

- **Cruce (Crossover):** Cruce Uniforme que intercambia **bloques completos de asignaciones de habilidad** entre padres. Esto garantiza la **factibilidad** estructural de la descendencia.
- **Mutación:** Se selecciona una habilidad k y su asignación se **reconstruye** completamente usando el **RGCH**, actuando como un movimiento aleatorio dentro del vecindario N^1 .

Algoritmos de Búsqueda Comparados

Se evaluaron siete algoritmos bajo un presupuesto de **Evaluaciones de Función (NFE)** escalado para garantizar una comparación justa.

Clase de Algoritmo	Estrategia	Algoritmos
Líneas Base Determinista	Construcción no estocástica	Greedy (DG)
Líneas Base Estocástica	Generación de soluciones al azar (Explotación de RGCH)	Random Search (RS)
Búsqueda Focalizada	Explotación de N^1	Hill Climbing (HC), Tabu Search (TS)
Nuestro Enfoque	Explotación Persistente de N^1	Stochastic Local Search (SLS)
Metaheurísticas	Diversificación y Población	Genetic Algorithm (GA), Variable Neighborhood Search (VNS)

Diseño y Metodología

- **Diseño:** 30 ejecuciones independientes por algoritmo (Muestras Pareadas).
- **Métrica:** Eficiencia media (E_{mean}).

Pruebas de Normalidad

- **Razón:** Los resultados acotados ($E \in [0, 1]$) producen **asimetría negativa (efecto de techo)**.
- **Verificación:** Prueba de **Shapiro-Wilk**.

Prueba de Hipótesis Principal

- **Test:** **Wilcoxon Signed-Rank Test** (no paramétrico).
- **Hipótesis Nula (H_0):** No hay diferencia significativa en la calidad de la solución.

Interpretación del p -valor ($\alpha = 0.05$)

Condición	Veredicto	Implicación Práctica
$p < 0.05$	Rechazar H_0	Superioridad Estadística
$p \geq 0.05$	No hay evidencia para rechazar H_0	Rendimiento Comparable

Configuración Experimental

Escalas de Instancias

Escala	Personas ($ \mathcal{H} $)	Proyectos ($ \mathcal{P} $)	Habilidades ($ \mathcal{K} $)	Presupuesto (NFE)
Pequeña	20	3	2	20,000
Mediana	100	10	10	50,000
Grande	200	20	20	100,000

Densidad sociométrica: 30% de relaciones positivas

Resultados: Resumen de Rendimiento

Pequeña (N=20)

Algorithm	Mean ± Std	Best	Time(s)
Greedy	0.694 ± 0.000	0.694	0.01
Hill Climbing	0.801 ± 0.028	0.855	0.01
Random Search	0.878 ± 0.011	0.912	8.50
Tabu Search	0.891 ± 0.009	0.912	4.99
Genetic Alg. (GA)	0.903 ± 0.015	0.932	11.77
VNS	0.907 ± 0.017	0.939	5.23
Stoch. LS (Ours)	0.896 ± 0.018	0.939	4.98

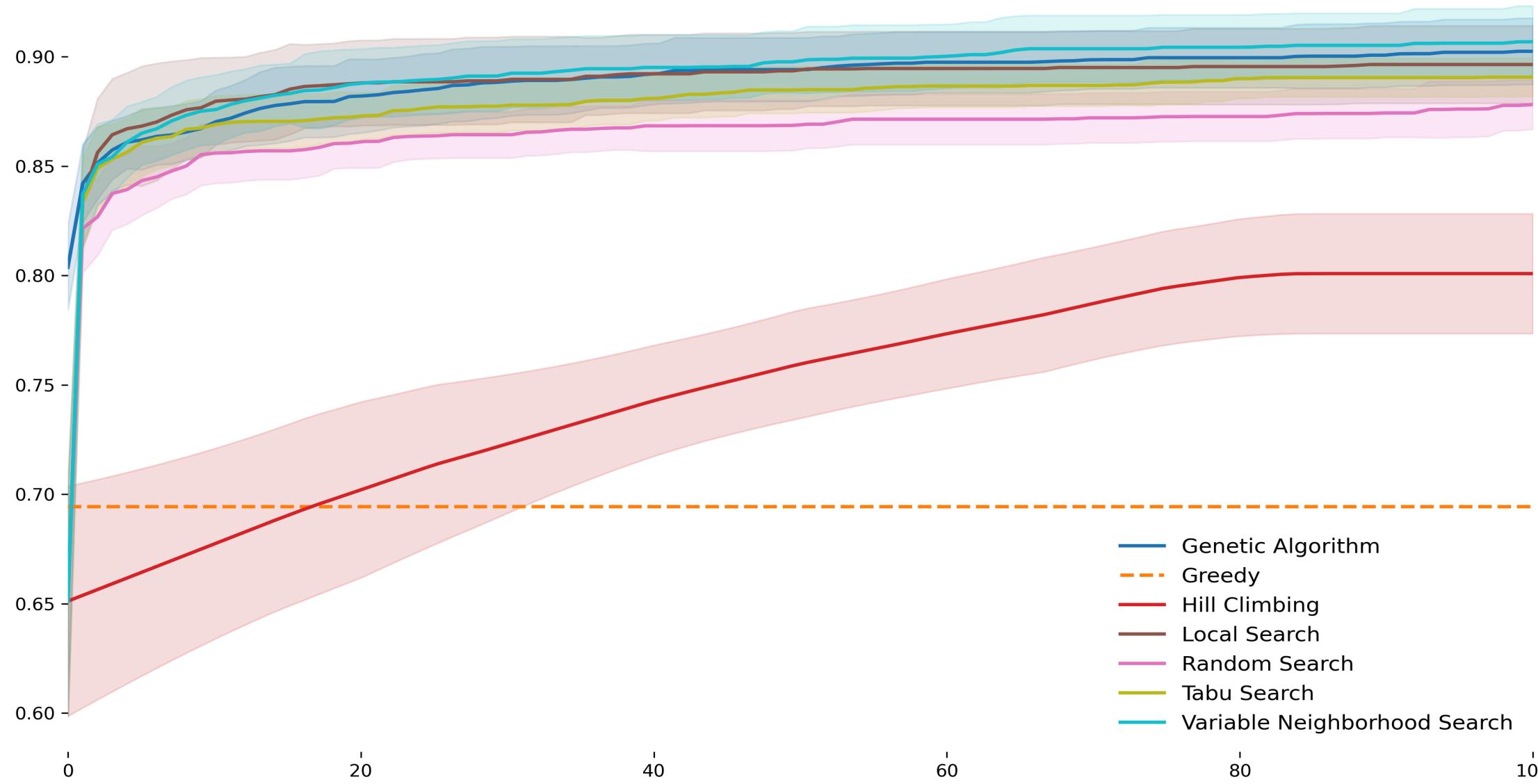
Mediana (N=100)

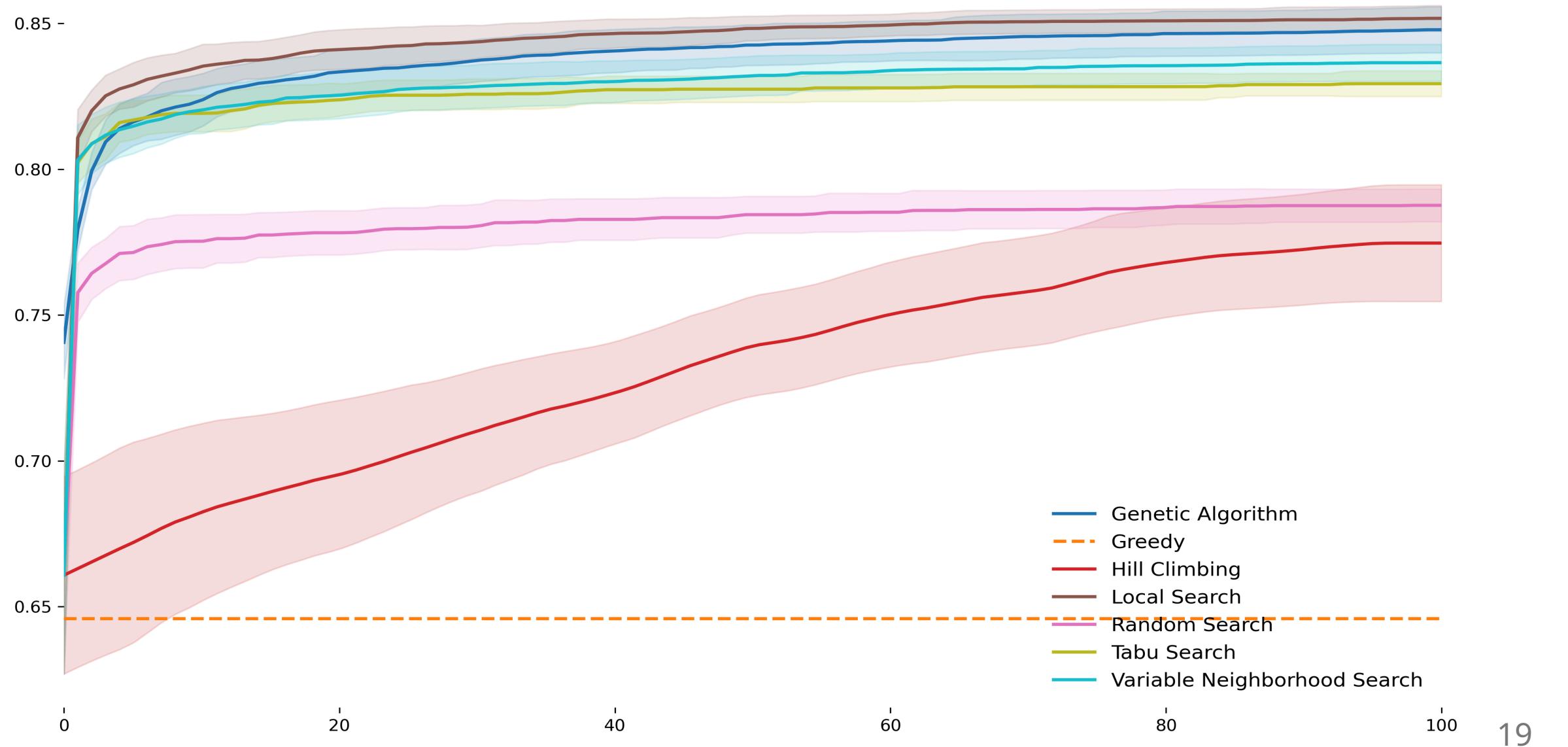
Algorithm	Mean ± Std	Best	Time(s)
Greedy	0.646 ± 0.000	0.646	0.01
Hill Climbing	0.775 ± 0.020	0.812	0.32
Random Search	0.788 ± 0.006	0.800	1129.1
Tabu Search	0.829 ± 0.004	0.841	149.6
Genetic Alg. (GA)	0.848 ± 0.008	0.862	250.6
VNS	0.837 ± 0.006	0.849	163.0
Stoch. LS (Ours)	0.852 ± 0.004	0.861	148.1

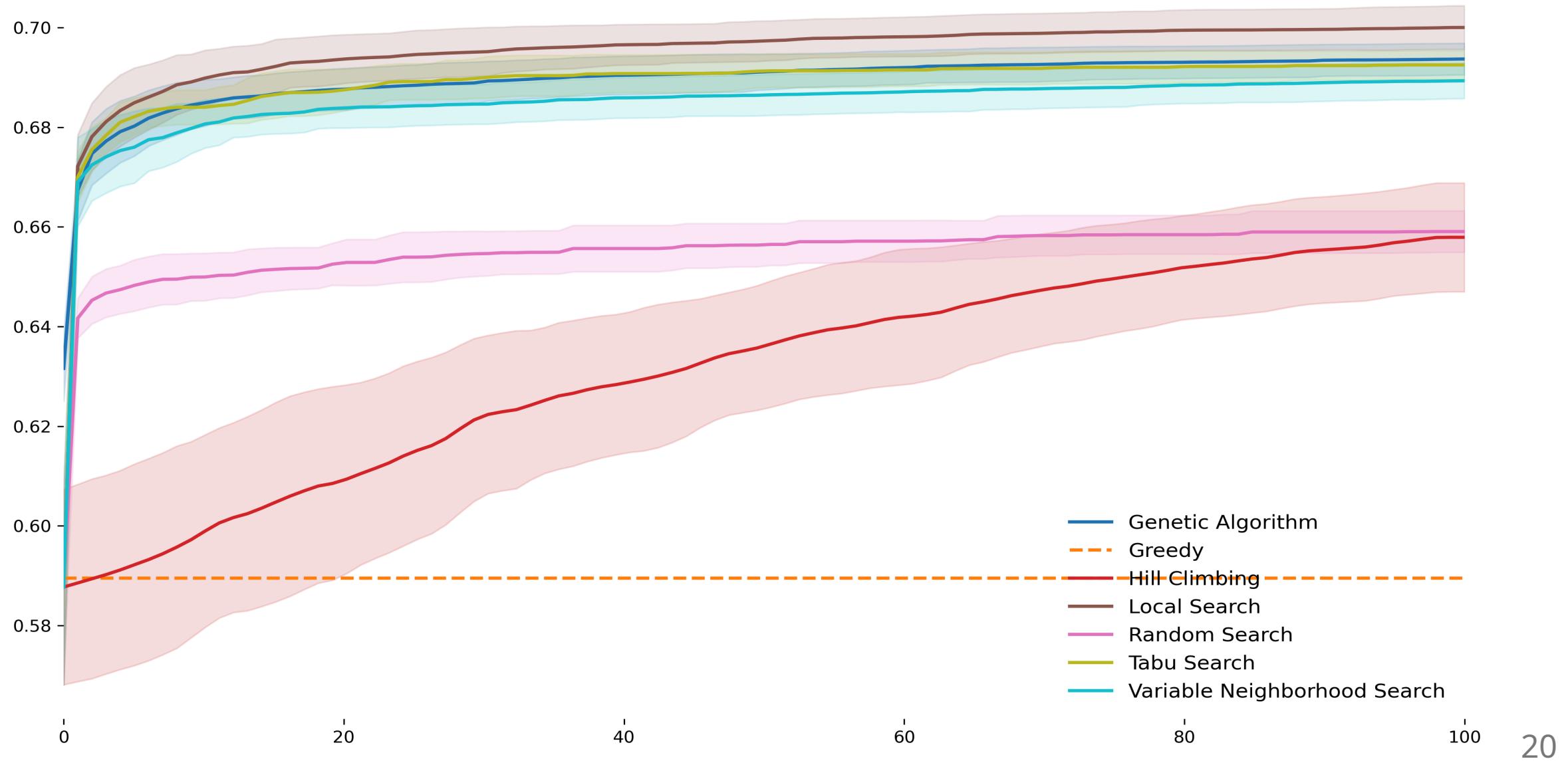
Grande (N=200)

Algorithm	Mean ± Std	Best	Time(s)
Greedy	0.590 ± 0.000	0.590	0.01
Hill Climbing	0.658 ± 0.011	0.672	3.68
Random Search	0.659 ± 0.004	0.669	16692.9
Tabu Search	0.692 ± 0.003	0.699	1104.7
Genetic Alg. (GA)	0.694 ± 0.003	0.700	2036.3
VNS	0.689 ± 0.004	0.697	1284.5
Stoch. LS (Ours)	0.700 ± 0.004	0.708	1103.4

Comportamiento de Convergencia







Test Estadístico ($N = 200$)

Prueba de Normalidad (Shapiro-Wilk)

Para la instancia Grande ($N = 200$), verificamos la distribución de las 30 ejecuciones:

Algoritmo	p-valor Shapiro-Wilk	Veredicto
SLS	0.098	No Rechaza H_0 (Normal)
GA	0.002	Rechaza H_0 (No Normal)
VNS	0.011	Rechaza H_0 (No Normal)

Implicación: Dado que el GA y el VNS no cumplen la condición de normalidad, se requiere el uso de pruebas no paramétricas para asegurar la validez de la comparación.

Análisis de Significación (Wilcoxon Signed-Rank Test)

Se compara el **SLS (0.700)** contra sus principales competidores, utilizando un diseño de muestras pareadas.

Comparación (SLS vs.)	Diferencia Media (d)	Desv. Estándar (σ)	p-valor	Veredicto
Genetic Algorithm (GA)	0.0063	0.0019	1.17×10^{-6}	Superior
Variable Neighborhood Search (VNS)	0.011	0.0028	1.86×10^{-9}	Superior
Tabu Search (TS)	0.008	0.0019	8.20×10^{-8}	Superior

El rendimiento superior de SLS en el caso de Gran Escala es estadísticamente significativo.

El p -valor extremadamente bajo (a pesar de la pequeña diferencia absoluta) se debe a la **alta estabilidad** de los resultados ($\sigma \approx 0.004$) y a la potencia del diseño de muestras pareadas.

Análisis de la Dominancia de SLS

¿Por qué SLS supera a Metaheurísticas complejas?

El resultado es contraintuitivo: un algoritmo simple supera a VNS y GA en alta dimensión.

Factor	GA / VNS	SLS (Explotación)
Enfoque	Diversificación (Búsqueda global)	Explotación Intensiva (Búsqueda focalizada)
Search Landscape	Altamente restringido y decomponible.	Favorece movimientos pequeños y precisos.
Eficiencia	La sobrecarga de la gestión de población (GA) o el <i>shaking</i> agresivo (VNS) es costosa.	La persistencia en la explotación del vecindario N^1 es clave para una convergencia profunda.

Conclusiones y Trabajo Futuro

Conclusiones

1. Se validó con éxito un **marco heurístico completo** ($RGCH + N^1$) para el MTFP, eliminando la dependencia de solvers exactos.
2. **Stochastic Local Search (SLS)** es el algoritmo más **robusto y escalable** para el MTFP de alta dimensión.
3. Se demostró que la **explotación intensiva** es más efectiva que la diversificación agresiva o poblacional.