

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SAN JOAQUÍN - CHILE



“DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH
TRAJECTORY NETWORKS ADAPTADO AL
SINTONIZADOR IRACE”

PABLO IGNACIO ESTOBAR FERNÁNDEZ

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Nicolás Rojas Morales
Profesor Correferente: Leslie Pérez Cáceres

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR IRACE

Nombre del candidato(a): Pablo Ignacio Estobar Fernández

Carrera / Grado: Ingeniería Civil Informática

Campus: Santiago San Joaquín; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Nicolás Rojas Morales, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☐ El trabajo **NO** contiene información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☒ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☒ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

Se espera que la propuesta y resultados expuestos en esta memoria sean publicados en un artículo de revista científica.

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 13/08/25

; Firma:

Estudiante o Candidato(a):

Fecha: 12/08/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedico este trabajo a mi familia, cuyo apoyo ha sido incondicional a lo largo de todo mi proceso universitario. En especial, a mis padres, quienes con esfuerzo diario me han brindado la oportunidad de estudiar y desarrollarme en las mejores condiciones, acompañándome siempre con cariño, dedicación y confianza.

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que me han apoyado, tanto en el desarrollo de este trabajo como a lo largo de mi vida personal y académica.

A mis padres, por su dedicación incondicional y por haberme dado la posibilidad de estudiar en condiciones adecuadas. A mi madre, por estar siempre dispuesta a escucharme, acompañarme y ayudarme a resolver mis problemas, especialmente en los momentos más exigentes. A mi padre, por su constante disposición para ayudarme en todo lo que he necesitado, asumiendo incluso aquellas tareas que no he podido realizar debido a mis responsabilidades.

A mi hermana, por su apoyo incondicional y por estar siempre presente, ayudándome de muchas formas a sobrellevar los periodos más intensos del proceso. A mi abuela, por su cercanía, sus consejos y su disposición constante para ayudar.

También quiero agradecer a mis amigos, quienes me han acompañado durante toda esta etapa. Su apoyo, ya sea estudiando juntos, compartiendo momentos en la universidad o simplemente desconectando en actividades de ocio, ha sido fundamental para mantener el equilibrio durante este camino.

Un agradecimiento muy especial a mi profesor guía, Nicolás Rojas, por su constante motivación, por haberme impulsado a participar como ayudante en el departamento y por acompañar de forma cercana y comprometida el desarrollo de este trabajo. Su orientación, dedicación y confianza han sido clave para que pudiera desarrollar este trabajo, y para crecer tanto académica como personalmente durante este proceso.

Asimismo, agradezco a quienes colaboraron directamente en la elaboración de este trabajo, que surgió como una investigación puntual y que fue tomando forma hasta convertirse en un proyecto de mayor alcance. Gracias por proporcionarme información, recursos y valiosos aportes que permitieron llevar esta memoria a buen término.

Finalmente, a todas las personas que no he mencionado explícitamente pero que de una u otra forma contribuyeron en este proceso: muchas gracias.

RESUMEN

Resumen— La configuración adecuada de parámetros en algoritmos metaheurísticos es esencial para asegurar su buen rendimiento, dado que estos algoritmos suelen operar en espacios de búsqueda complejos y con alto grado de variabilidad. En este contexto, esta memoria presenta STN-i, un modelo de *Search Trajectory Networks* adaptado al configurador automático *irace*, cuyo propósito es representar, visualizar y analizar estructuralmente el proceso de sintonización de parámetros. La propuesta permite modelar configuraciones, relaciones de generación e información estructural, e incluye además una variante denominada merged STN-i para el análisis comparativo entre múltiples ejecuciones. La validación se lleva a cabo sobre tres algoritmos metaheurísticos (MMASQAP, ACOTSP y PSO-X), cinco conjuntos de instancias y distintas condiciones de presupuesto y uso del mecanismo de *soft-restart*, generando 60 redes STN-i y 30 merged STN-i, construidas a partir de tres discretizaciones del espacio paramétrico. A través de métricas estructurales y visualizaciones mediante *layouts* de fuerza, se identifican patrones relevantes como trayectorias dominantes, zonas de convergencia y divergencia, y regiones compartidas entre ejecuciones. Los resultados evidencian que STN-i entrega una representación detallada y explicativa del comportamiento de *irace*, fortaleciendo su análisis, interpretación y comparación estructural en contextos de configuración automática.

Palabras Clave— *irace*, Search Trajectory Networks, Metaheurísticas, Sintonización de parámetros

ABSTRACT

Abstract— The proper configuration of parameters in metaheuristic algorithms is essential to ensure their effective performance, as these algorithms typically operate over complex and highly variable search spaces. In this context, this thesis presents STN-i, a *Search Trajectory Network* model adapted to the automatic configurator *irace*, aimed at representing, visualizing, and structurally analyzing the parameter tuning process. The proposed model captures configurations, generative relationships, and structural information, and includes a variant called merged STN-i, designed for comparative analysis across multiple runs. Validation is carried out using three metaheuristic algorithms (MMASQAP, ACOTSP, and PSO-X), five instance sets, and varying conditions of computational budget and use of the *soft-restart* mechanism, resulting in 60 STN-i networks and 30 merged STN-i networks

built from three levels of parameter space discretization. Using structural metrics and force-directed visualizations, the model identifies relevant patterns such as dominant trajectories, convergence and divergence zones, and shared regions across executions. The results show that STN-i provides a detailed and explanatory representation of *irace*'s behavior, enhancing its structural analysis, interpretability, and comparative insight in the context of automatic configuration.

Keywords— *irace*, Search Trajectory Networks, Metaheuristics, Automatic Parameter Configuration

GLOSARIO

Problema de optimización: Tarea de encontrar la mejor solución dentro de un conjunto de alternativas factibles, minimizando o maximizando una función objetivo, sujeta a restricciones.

Problema NP-hard / NP-complete: Clasificaciones de complejidad computacional asociadas a la dificultad de resolver o verificar eficientemente problemas de decisión u optimización.

Técnicas completas: Métodos de resolución que garantizan encontrar la solución óptima (si existe), como backtracking o branch and bound, pero que pueden ser computacionalmente inviables.

Técnicas incompletas: Métodos que no garantizan la solución óptima, pero ofrecen soluciones de buena calidad en tiempos razonables. Incluyen heurísticas y metaheurísticas.

Heurística: Técnica de resolución que guía la búsqueda usando conocimiento específico del problema. Usualmente eficiente pero limitada a contextos particulares.

Metaheurística: Enfoque generalizado de optimización que integra heurísticas para guiar la búsqueda de soluciones en dominios complejos. Ejemplos incluyen GA, PSO, ACO, SA y TS.

Exploración (diversificación): Capacidad de un algoritmo para buscar en nuevas áreas del espacio de soluciones.

Explotación (intensificación): Capacidad de un algoritmo para mejorar las soluciones en áreas ya identificadas como prometedoras.

Parámetros: Valores internos que controlan el comportamiento de un algoritmo (por ejemplo, temperatura inicial en SA o probabilidad de cruce en GA).

Parameter Setting Problem: Problema de encontrar una configuración de parámetros adecuada para un algoritmo dado un contexto particular.

Control de parámetros: Enfoque que ajusta dinámicamente los valores de parámetros durante la ejecución del algoritmo.

Configuración automática de parámetros: Estrategia que ajusta parámetros antes de ejecutar el algoritmo, usando técnicas como búsqueda aleatoria, evolución o racing.

irace: Configurador automático de algoritmos basado en técnicas de racing, que selecciona configuraciones prometedoras mediante comparaciones estadísticas y procesos iterativos de evaluación.

Soft-restart: Mecanismo de irace que reinserta parcialmente diversidad en la búsqueda del sintonizador cuando hay estancamiento o convergencia prematura.

Search Trajectory Networks (STN): Modelo de red que representa trayectorias de búsqueda de algoritmos, mediante nodos que representan soluciones y arcos que representan transiciones o relaciones entre ellas.

STN-i: Adaptación del modelo STN al configurador irace, incorporando nodos con clasificación estructural, arcos de herencia (parent-edges) y trayectorias paralelas por iteración.

Merged STN-i: Variante del modelo STN-i que fusiona múltiples ejecuciones para analizar comparativamente regiones de convergencia o divergencia estructural.

Parent-edge: Arco dirigido en una STN-i que conecta una configuración con su(s) configuración(es) élite generadora(s).

Locación: Representación discretizada de una configuración del espacio de parámetros. Su nivel de granularidad afecta la estructura de las STN-i. Discretiza-

ción: Representaciones del espacio de parámetros con cierta granularidad mediante alguna locación. Nodos elite/regulares/de mejor calidad: Clasificación estructural de nodos en las STN-i, según su rol en el proceso de búsqueda y rendimiento relativo. Nodos shared/network: Clasificación adicional en merged STN-i que distingue nodos compartidos entre redes y nodos exclusivos de una red particular. Fruchterman-Reingold: Algoritmo de disposición gráfica basado en fuerzas, utilizado para visualizar redes de forma legible y estructurada. Métricas estructurales: Conjunto de indicadores cuantitativos diseñados para caracterizar las STN-i, como número de nodos, proporciones por tipo, longitud de trayectorias o conectividad entrante.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	VI
ÍNDICE DE FIGURAS	X
ÍNDICE DE TABLAS	XIV
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	3
CAPÍTULO 2: MARCO CONCEPTUAL	7
2.1 Problemas de optimización	7
2.1.1 Travelling Salesman Problem (TSP)	9
2.1.2 Quadratic Assignment Problem (QAP)	9
2.2 Técnicas Completas	10
2.3 Técnicas Incompletas	11
2.3.1 Heurísticas	12
2.3.2 Metaheurísticas	12
2.3.3 Dependencia de parámetros en técnicas incompletas	20
2.4 Parameter Setting Problem	21
2.4.1 Control de parámetros	22
2.4.2 Configuración automática de parámetros	23
2.5 irace	25
2.6 Search Trajectory Networks	29
2.6.1 Definición del modelo	30
2.6.2 Métricas	33
2.6.3 Aplicaciones de STNs en la literatura	34
2.7 Resumen	38
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	39
3.1 Definición del modelo propuesto de STN adaptado a irace (STN-i)	39
3.1.1 Merged STN-i	42
3.1.2 Métricas de los modelos STN-i y merged STN-i	44
3.2 Algoritmo <i>irace-STN-i</i> : Extracción e interpretación de trayectorias de irace	47
3.2.1 Etapa 1: Datos de entrada	48
3.2.2 Etapa 2: Lectura de datos	49
3.2.3 Etapa 3: Construcción de trayectorias	49
3.2.4 Etapa 4: Generación de archivo de salida	50
3.3 Conclusiones	51

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	53
4.1 Escenario experimental	53
4.1.1 Algoritmos objetivo	54
4.1.2 Instancias	57
4.1.3 Locaciones	58
4.1.4 Hiperparámetros de <i>irace</i>	60
4.1.5 Metodología de validación de la solución	61
4.2 Análisis y resultados	62
4.2.1 Resultados de MMASQAP (QAP-60)	63
4.2.2 Análisis de MMASQAP (QAP-60)	70
4.2.3 Resultados de ACOTSP (TSP-2000)	73
4.2.4 Análisis de ACOTSP (TSP-2000)	80
4.2.5 Resultados de PSO-X (PSO-Mix)	83
4.2.6 Análisis de PSO-X (PSO-Mix)	90
4.2.7 Resultados de PSO-X (PSO-Mul)	93
4.2.8 Análisis de PSO-X (PSO-Mul)	100
4.2.9 Resultados de PSO-X (PSO-Uni)	103
4.2.10 Análisis de PSO-X (PSO-Uni)	110
4.2.11 Análisis general	113
4.3 Conclusiones	117
CAPÍTULO 5: CONCLUSIONES	119
5.1 Trabajo futuro	123
REFERENCIAS BIBLIOGRÁFICAS	126

ÍNDICE DE FIGURAS

1	Árbol del problema.	5
2	Representación del comportamiento colectivo observado en bandadas de aves y cardúmenes de peces.	17
3	Esquema general del funcionamiento de <i>irace</i>	26
4	Ejemplo de una STN.	30
5	Ejemplo de una LON.	30
6	Ejemplo de merged STN.	32
7	Merged STN construida para la instancia $p9c9$ del CBSP, combinando trayectorias del MA clásico y el DMAB+MA. Fuente: [Narvaez-Teran <i>et al.</i> , 2021]. . .	34
8	Merged STNs para el problema UF8. En la izquierda el algoritmo MOEA/D y a la derecha el NSGA-II respectivamente.	36
9	Merged STN para la instancia 4-20 con precisión p_2 y $s = 10$, combinando trayectorias de ParamILS y Evoca.	37
10	Ejemplo de STN-i generada a partir de múltiples ejecuciones independientes de <i>irace</i>	41
11	Ejemplo del modelo merged STN-i construido a partir de dos redes STN-i generadas de forma independiente sobre un mismo algoritmo, variando únicamente las condiciones de los hiperparámetros de <i>irace</i>	43
12	Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	63
13	Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	64
14	Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	65
15	Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	66

16	Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia QAP-60 con el algoritmo MMASQAP. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	67
17	Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia QAP-60 con el algoritmo MMASQAP. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	68
18	Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia QAP-60 del algoritmo MMASQAP.	70
19	Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	73
20	Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	74
21	Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	75
22	Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	76
23	Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia TSP-2000 con el algoritmo ACOTSP. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	77

24	Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia <i>TSP-2000</i> con el algoritmo ACOTSP. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	78
25	Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia <i>TSP-2000</i> del algoritmo ACOTSP.	80
26	Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia <i>PSO-Mix</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	83
27	Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia <i>PSO-Mix</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	84
28	Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia <i>PSO-Mix</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	85
29	Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia <i>PSO-Mix</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1, L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	86
30	Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia <i>PSO-Mix</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	87
31	Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia <i>PSO-Mix</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1, L_2, L_3) , con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	88
32	Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia <i>PSO-Mix</i> del algoritmo PSO-X.	90

33	Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia <i>PSO-Mul</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	93
34	Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia <i>PSO-Mul</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	94
35	Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia <i>PSO-Mul</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	95
36	Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia <i>PSO-Mul</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	96
37	Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia <i>PSO-Mul</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	97
38	Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia <i>PSO-Mul</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	98
39	Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia <i>PSO-Mul</i> del algoritmo PSO-X.	100
40	Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia <i>PSO-Uni</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	103
41	Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia <i>PSO-Uni</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	104

42	Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia <i>PSO-Uni</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	105
43	Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia <i>PSO-Uni</i> utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.	106
44	Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia <i>PSO-Uni</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	107
45	Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia <i>PSO-Uni</i> con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos <i>network-regular</i> ni <i>shared-regular</i> provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.	108
46	Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia <i>PSO-Uni</i> del algoritmo PSO-X.	110
47	Proporción promedio entre nodos y configuraciones en las STN-i, diferenciando por combinación de algoritmo objetivo, conjunto de instancias y nivel de presupuesto.	115
48	Representación de las tendencias de las STN-i para las métricas más relevantes, agrupadas por cada nivel de presupuesto computacional.	115
49	Representación de las tendencias de las merged STN-i para las métricas más relevantes, agrupadas por cada nivel de presupuesto computacional.	116

ÍNDICE DE TABLAS

1	Parámetros que serán optimizados para MMASQAP.	54
2	Parámetros a sintonizar para ACOTSP.	55
3	Parámetros a sintonizar para PSO-X.	56
4	Discretización de parámetros por locación en MMASQAP.	59
5	Discretización de parámetros por locación en ACOTSP.	59
6	Discretización de parámetros por locación en PSO-X.	60
7	Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de <i>irace</i> para cada uno de los escenarios del algoritmo MMAS-QAP sobre la instancia QAP-60.	69

8	Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo MMASQAP en la instancia QAP-60.	69
9	Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia QAP-60 del algoritmo MMASQAP.	69
10	Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo ACOTSP sobre la instancia TSP-2000.	79
11	Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo ACOTSP en la instancia TSP-2000.	79
12	Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia TSP-2000 del algoritmo ACOTSP.	79
13	Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia PSO-Mix.	89
14	Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia PSO-Mix.	89
15	Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia PSO-Mix del algoritmo PSO-X.	89
16	Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia PSO-Mul.	99
17	Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia PSO-Mul.	99
18	Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia PSO-Mul del algoritmo PSO-X.	99
19	Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia PSO-Uni.	109
20	Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia PSO-Uni.	109
21	Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia PSO-Uni del algoritmo PSO-X.	109

INTRODUCCIÓN

La configuración de algoritmos es una etapa crítica en el diseño y aplicación de métodos de optimización, especialmente en entornos donde se requiere adaptar el comportamiento de un algoritmo a características particulares del problema. Este desafío cobra especial importancia en el caso de los algoritmos metaheurísticos, ampliamente utilizados para resolver problemas complejos y no estructurados, cuya eficacia depende en gran medida de la adecuada selección de múltiples parámetros internos. Una mala configuración puede resultar en bajo rendimiento, convergencia prematura o falta de robustez, mientras que una buena elección puede potenciar significativamente sus capacidades de búsqueda.

Ante esta necesidad, ha emergido el campo de la configuración automática de algoritmos, el cual busca encontrar de forma sistemática y eficiente una combinación adecuada de parámetros, sin intervención manual ni necesidad de modificar la lógica del algoritmo original. En este contexto, el algoritmo *irace* se ha posicionado como una de las herramientas más populares, gracias a su estrategia basada en técnicas de *racing*, que guían el proceso de selección de configuraciones mediante comparaciones estadísticas sucesivas. No obstante, *irace* posee una dinámica interna compleja: genera múltiples configuraciones en paralelo a partir de un conjunto élite, evalúa iterativamente bajo criterios adaptativos y puede aplicar mecanismos como el *soft-restart*, lo cual dificulta entender su comportamiento desde una perspectiva estructural. Esta opacidad plantea un problema para quienes buscan no solo ejecutar procesos de configuración, sino también comprender cómo y por qué ciertas configuraciones son seleccionadas o descartadas.

En este contexto, el problema que se aborda en esta memoria es la falta de herramientas que permitan interpretar y analizar el comportamiento de *irace* desde una perspectiva estructural, visual y comparativa. Se plantea entonces el desarrollo de un modelo que permita representar las trayectorias exploradas durante la sintonización, identificando relaciones entre configuraciones, evolución de la búsqueda y patrones recurrentes o divergentes dentro del espacio paramétrico.

La solución propuesta consiste en el modelo STN-i (*Search Trajectory Network* adaptado a *irace*), una adaptación de las Search Trajectory Networks tradicionalmente usadas para analizar algoritmos secuenciales, extendidas aquí para capturar la semántica propia de *irace*. Este modelo incorpora elementos específicos como arcos de herencia entre configuraciones (*parent-edges*), categorización estructural de nodos (por rol y calidad relativa), trayectorias paralelas generadas por iteraciones, y permite representar procesos con múltiples fuentes de generación. Además, se propone una variante denominada merged STN-i, orientada al análisis comparativo de múltiples ejecuciones bajo diferentes condiciones experimentales, permitiendo observar zonas de convergencia, divergencia o superposición en el espacio de búsqueda.

Para validar este modelo, se diseña un estudio experimental que considera tres algoritmos

metaheurísticos de distinta naturaleza (MMASQAP, ACOTSP y PSO-X), cinco conjuntos de instancias, dos niveles de presupuesto computacional y la activación opcional del mecanismo de *soft-restart*. A partir de los resultados obtenidos mediante ejecuciones controladas de *irace*, se construyen 60 redes STN-i y 30 merged STN-i, aplicando tres niveles de discretización del espacio de parámetros. Estas redes son posteriormente analizadas mediante métricas estructurales definidas sobre el modelo y visualizadas mediante algoritmos de disposición por fuerza, lo cual permite estudiar en detalle el comportamiento exploratorio y estructural de cada escenario.

Este documento está organizado de la siguiente manera. En el capítulo 1 se presenta la definición del problema que da origen a esta memoria, junto con su motivación, alcance, y los objetivos generales y específicos del trabajo. A continuación, el capítulo 2 entrega el marco conceptual necesario para comprender el contexto técnico del problema abordado. Se introduce el dominio general de la optimización, las características de los problemas que la motivan, y las distintas técnicas de resolución existentes, incluyendo métodos completos, heurísticos e incompletos. Se profundiza en las metaheurísticas como técnicas clave para enfrentar problemas complejos, y se revisa la problemática asociada a la configuración de sus parámetros, diferenciando entre control y sintonización automática. En este contexto, se presenta el sintonizador *irace*, sus componentes principales y funcionamiento, así como también el concepto de *Search Trajectory Networks* (STNs), abordando sus fundamentos, usos en la literatura y aplicaciones previas en visualización y análisis estructural de algoritmos. Luego, el capítulo 3 describe la propuesta desarrollada en esta memoria, introduciendo el modelo STN-i, sus componentes clave, variantes y las consideraciones metodológicas adoptadas para su implementación. Posteriormente, en el capítulo 4 se describe el diseño experimental desarrollado para validar el modelo propuesto, incluyendo la selección de algoritmos metaheurísticos, conjuntos de instancias, parámetros evaluados y condiciones experimentales. En la sección 4.2, se presentan y analizan en detalle los resultados obtenidos, aplicando métricas estructurales y visualizaciones que permiten estudiar el comportamiento del proceso de configuración. Finalmente, el capítulo 5 expone las conclusiones generales del trabajo realizado, junto con una discusión de posibles líneas de investigación futura para ampliar y profundizar el modelo propuesto.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

Los problemas de optimización buscan determinar la mejor solución entre un conjunto finito de alternativas factibles, maximizando o minimizando una función objetivo definida sobre un espacio de búsqueda [Griva *et al.*, 2008, Schrijver, 2003]. La resolución de estos problemas ha dado origen a una amplia gama de métodos, que se distinguen principalmente entre técnicas completas e incompletas [Garey y Johnson, 1979, Russell y Norvig, 2020]. Las primeras garantizan encontrar la solución óptima, aunque suelen ser inviables computacionalmente para problemas de gran escala. Por ello, han surgido enfoques incompletos como las heurísticas y metaheurísticas, los cuales sacrifican la garantía de obtener soluciones óptimas a cambio de una mayor eficiencia, permitiendo obtener soluciones de buena calidad en tiempos razonables [Glover, 1986, Blum y Roli, 2003, Pearl, 1984].

Dentro de este conjunto de enfoques incompletos, las metaheurísticas destacan como técnicas generales y versátiles para abordar problemas de optimización complejos. En particular, permiten encontrar soluciones de buena calidad en espacios de búsqueda difíciles, manteniendo un equilibrio entre eficiencia computacional y resultados aceptables. Estas técnicas implementan heurísticas auxiliares para guiar la exploración del espacio de soluciones de forma efectiva. A diferencia de las heurísticas tradicionales, que suelen estar diseñadas para dominios específicos, las metaheurísticas están concebidas para aplicarse de manera general sobre diversos tipos de problemas [Abdel-Basset *et al.*, 2018]. Su funcionamiento se apoya en dos procesos fundamentales: exploración (diversificación)¹ y explotación (intensificación)². Algunas estrategias priorizan uno de estos aspectos, mientras que otras buscan un equilibrio dinámico entre ambos para optimizar el rendimiento global. Entre las metaheurísticas más ampliamente utilizadas se encuentran Genetic Algorithms (GA) [Holland, 1992], Particle Swarm Optimization (PSO) [Kennedy y Eberhart, 1995], Ant Colony Optimization (ACO) [Dorigo *et al.*, 1996], Simulated Annealing (SA) [Kirkpatrick *et al.*, 1983] y Tabu Search (TS) [Glover, 1990].

El desempeño de las metaheurísticas depende de los valores de sus parámetros, y es común que un conjunto de valores adecuado para un problema no lo sea para otro, ni siquiera para distintas instancias del mismo problema. Por ejemplo, en SA, la temperatura inicial determina el grado de aceptación de soluciones que empeoran la actual, mientras que en los GA, la probabilidad de *cruce*³ define la tendencia en la generación de nuevas soluciones. Estos parámetros son fundamentales para el funcionamiento de los algoritmos, ya que influyen directamente en su rendimiento y en la calidad de las soluciones obtenidas. Encontrar la configuración óptima consiste en el *Parameter Setting Problem*, el cual es un problema

¹Capacidad de un algoritmo para buscar en nuevas áreas del espacio de soluciones.

²Capacidad de un algoritmo para mejorar las soluciones en áreas ya identificadas.

³Operador utilizado en algoritmos poblacionales que genera nuevas soluciones combinando partes de dos o más soluciones existentes.

complejo pues requiere una considerable inversión de tiempo y esfuerzo para garantizar su eficacia en distintas instancias del problema [Montero *et al.*, 2014].

Por esta razón, se han desarrollado dos enfoques principales para abordar este desafío: el control de parámetros y la configuración automática de parámetros. El control de parámetros ajusta dinámicamente los valores durante la ejecución del algoritmo, permitiendo que estos se adapten a diferentes situaciones en tiempo real. En contraste, la configuración automática de parámetros —también denominada sintonización de parámetros— se enfoca en encontrar una configuración adecuada antes de la ejecución, manteniendo los valores fijos a lo largo de la ejecución del algoritmo. Dentro de este último enfoque, se han propuesto diversas técnicas, entre las cuales se destacan ParamILS [Hutter *et al.*, 2009], EVOCA [Riff y Montero, 2013], F-RACE [Birattari *et al.*, 2010], *irace* [López-Ibáñez *et al.*, 2016], SMAC [Hutter *et al.*, 2011] y GGA++ [Ansótegui *et al.*, 2015], cada una con estrategias particulares para optimizar el rendimiento de los algoritmos.

En este trabajo se utilizará el sintonizador *irace*, una de las técnicas de configuración automática de parámetros más reconocidas por su eficiencia y capacidad de adaptación en diversos contextos de optimización. *irace* ha demostrado un desempeño competitivo en múltiples estudios comparativos, destacándose especialmente en problemas complejos donde el equilibrio entre exploración y explotación resulta crítico [López-Ibáñez *et al.*, 2016].

irace es un algoritmo de configuración automática basado en técnicas de *racing* y evaluación estadística. Durante su ejecución, compara la calidad de múltiples configuraciones en distintas instancias del problema, eliminando estadísticamente aquellas de bajo desempeño y generando nuevas soluciones a partir de las mejores, denominadas élites. Este ciclo se repite de forma iterativa hasta agotar el presupuesto computacional o alcanzar una condición de término, como un número máximo de evaluaciones. Entre sus mejoras, *irace* incorpora la técnica de *soft-restart*⁴ para prevenir la convergencia prematura y equilibrar exploración y explotación [López-Ibáñez *et al.*, 2016].

A pesar de su eficacia, *irace* —al igual que otros sintonizadores— enfrenta diversos desafíos [Cáceres *et al.*, 2017]. La Figura 1 presenta el árbol del problema abordado en esta memoria, donde el problema central radica en la dificultad de analizar, estudiar y comprender el proceso de búsqueda realizado por *irace*. Esto se debe, entre otros factores, a la escasa claridad sobre las interrelaciones entre parámetros, lo que dificulta ajustes precisos, limita la identificación de oportunidades de mejora y restringe una optimización eficiente. Además, la exploración ineficiente del espacio de búsqueda, sumada a trayectorias extensas y altos requerimientos computacionales, contribuye al estancamiento del proceso. Estas condiciones incrementan considerablemente los recursos necesarios —como el tiempo— para completar cada iteración, reduciendo la eficiencia general de *irace* y limitando su capacidad para encontrar soluciones óptimas o adecuadas.

Por otro lado, una herramienta gráfica útil para analizar la tendencia de los algoritmos

⁴Método que ajusta el proceso de optimización sin reiniciar desde cero.

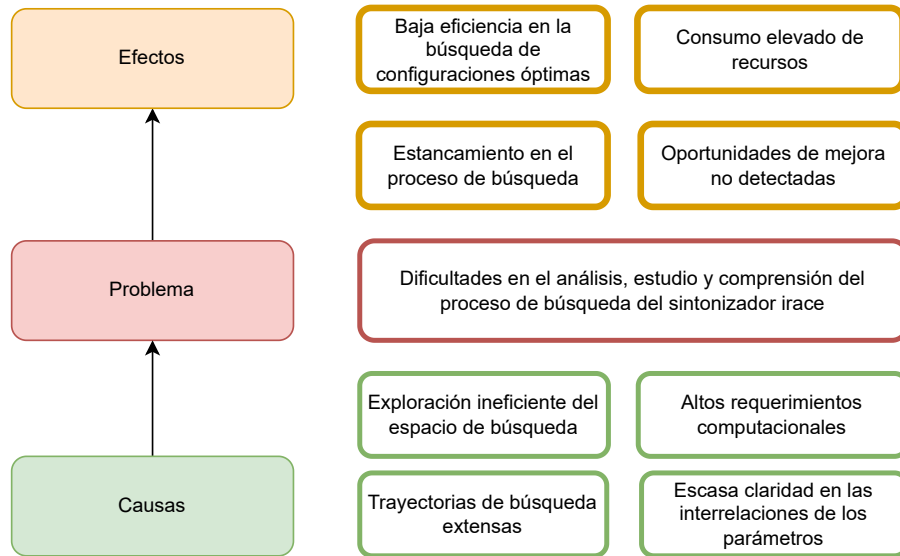


Figura 1: Árbol del problema.
Fuente: Elaboración propia.

metaheurísticos en el espacio de búsqueda son las Search Trajectory Networks (STNs) [Ochoa *et al.*, 2021]. Estas redes proporcionan una representación visual y cuantitativa de cómo los algoritmos exploran y explotan el espacio de parámetros. Una STN se define como un grafo dirigido $STN = G(N, E)$, donde N es el conjunto de nodos y E el conjunto de arcos. En este contexto:

- Un nodo representa una ubicación en el espacio de búsqueda en el proceso de optimización. Cada nodo modela una configuración de parámetros evaluada durante la búsqueda.
- Un arco conecta dos nodos visitados de manera consecutiva en el espacio de búsqueda, indicando la transición entre soluciones candidatas y delineando el trayecto del algoritmo a través del espacio de parámetros.

Las STNs permiten visualizar cómo los algoritmos balancean exploración y explotación, lo cual es clave para comprender su comportamiento en la optimización. En estudios recientes, se ha demostrado su utilidad para analizar los sintonizadores ParamILS y EVOCA [Riveros *et al.*, 2024]. Se observó que EVOCA genera redes desconectadas debido a su enfoque poblacional, mientras que ParamILS tiende a explorar configuraciones vecinas, logrando óptimos locales en cada ejecución. Sin embargo, al aumentar el número de semillas, ParamILS redujo la cantidad de nodos visitados, mientras que EVOCA fue más resiliente.

Aunque las STNs han mostrado ser útiles en estos análisis, irace aún no ha sido estudiado utilizando esta herramienta. Se han desarrollado trabajos que analizan aspectos específicos

del comportamiento de *irace*, como su estrategia de balance entre exploración y explotación [López-Ibáñez *et al.*, 2016], el impacto de sus parámetros internos en la eficacia del proceso de configuración [Pérez Cáceres *et al.*, 2014], y mejoras en su proceso de búsqueda mediante técnicas como el Adaptive Capping [Cáceres *et al.*, 2017]. Sin embargo, falta un estudio que permita visualizar y comprender de forma estructural las trayectorias de búsqueda que genera *irace*, limitando la capacidad de identificar patrones, oportunidades de mejora y optimizar su eficiencia, como se ilustra en el árbol del problema en la Figura 1.

Este trabajo propone utilizar STNs para evaluar el desempeño de *irace* en la sintonización de parámetros, con el fin de analizar su gestión de configuraciones en distintos contextos. El análisis ayudará a identificar el equilibrio entre exploración y explotación, y a descubrir posibles mejoras en la velocidad de convergencia y la calidad de las soluciones obtenidas.

En relación a lo mencionado, en este trabajo se propone como objetivo general el diseñar e implementar un modelo de Search Trajectory Networks adaptado al sintonizador de parámetros *irace* para mejorar el análisis y comprender su proceso de búsqueda aplicado a diversas metaheurísticas. Y los objetivos específicos son:

- Examinar el estado del arte sobre la sintonización de parámetros, las STN y el sintonizador *irace*, identificando enfoques existentes y oportunidades de mejora.
- Diseñar un modelo de STNs adaptado para representar y analizar las trayectorias de búsqueda generadas por *irace*.
- Implementar el modelo de STNs sintonizando algoritmos metaheurísticos con *irace*, permitiendo la visualización y análisis de las trayectorias de búsqueda.
- Analizar y evaluar las trayectorias de búsqueda generadas a través del modelo de STNs, identificando patrones, ventajas y desventajas de su proceso de búsqueda.

CAPÍTULO 2

MARCO CONCEPTUAL

En este capítulo se presentan los fundamentos teóricos necesarios para comprender la problemática abordada en esta memoria, así como las herramientas y enfoques utilizados en su análisis. Se inicia con una definición formal de los problemas de optimización y su clasificación, destacando aquellos de naturaleza combinatoria como el TSP y el QAP, que serán empleados como casos de estudio.

Posteriormente, se introducen las principales estrategias algorítmicas desarrolladas para abordar este tipo de problemas, diferenciando entre técnicas completas e incompletas. En particular, se profundiza en las técnicas incompletas —como las heurísticas y metaheurísticas—, detallando sus componentes clave, su comportamiento exploratorio y su dependencia de parámetros.

A continuación, se aborda el *Parameter Setting Problem*, destacando su impacto sobre la calidad de los resultados obtenidos por algoritmos metaheurísticos, y se presentan los enfoques desarrollados para resolverlo, tales como el control de parámetros y la configuración automática de parámetros. Entre estas últimas, se describe con detalle el funcionamiento del algoritmo *irace*, utilizado como herramienta principal en esta investigación para la configuración automática de parámetros.

Finalmente, se introduce el modelo de *Search Trajectory Networks* (STNs), el cual permite representar y analizar de manera estructurada las trayectorias generadas por algoritmos de búsqueda. Este modelo se propone como base metodológica para estudiar el comportamiento exploratorio de *irace*, proporcionando métricas topológicas e interpretaciones visuales del proceso de sintonización.

2.1. Problemas de optimización

Un problema de optimización consiste en determinar una solución $x^* \in \mathcal{X}$ que minimice (o maximice) una función objetivo $f : \mathcal{X} \rightarrow \mathbb{R}$ sobre un conjunto de decisiones factibles $\mathcal{X} \subseteq \mathbb{R}^n$, sujeto a un conjunto de restricciones explícitas e implícitas. La formulación general de un problema de optimización se expresa como:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{mín}} \quad & f(x) \\ \text{s.a.} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_j(x) = 0, \quad j = 1, \dots, p, \\ & x \in \mathcal{X}, \end{aligned} \tag{1}$$

donde g_i y h_j representan, respectivamente, las restricciones de desigualdad y de igualdad que delimitan el conjunto de soluciones factibles, mientras que $\mathcal{X}$ puede incluir restricciones adicionales de tipo estructural, tales como condiciones de integridad (por ejemplo, variables enteras o binarias), pertenencia a conjuntos discretos o restricciones de naturaleza combinatoria o lógica [Griva et al., 2008, Schrijver, 2003].

Una clasificación importante distingue a los problemas de optimización entre:

- Problemas de optimización continua, donde las variables de decisión toman valores en subconjuntos continuos de $\mathbb{R}^n$.
- Problemas de optimización discreta o combinatoria, donde al menos una variable está restringida a un conjunto finito o numerable (por ejemplo, enteros, permutaciones, subconjuntos). Estos problemas son particularmente relevantes en logística, asignación y planificación [Schrijver, 2003].

Desde el punto de vista computacional, los problemas de optimización se estudian bajo el marco de la teoría de la complejidad algorítmica, mediante problemas de decisión asociados. Un problema de decisión pertenece a la clase $\mathcal{P}$ si puede resolverse en tiempo polinomial respecto al tamaño de la entrada. En cambio, pertenece a la clase $\mathcal{NP}$ si, aunque no se conozca un algoritmo que lo resuelva en tiempo polinomial, toda solución candidata puede ser verificada en dicho tiempo. Muchos problemas de optimización combinatoria tienen versiones de decisión en $\mathcal{NP}$, y determinar si también están en $\mathcal{P}$ constituye uno de los grandes desafíos en informática teórica [Garey y Johnson, 1979].

En este contexto, un problema de decisión es $\mathcal{NP}$ -complete si está en $\mathcal{NP}$ y es al menos tan difícil como cualquier otro problema en esa clase, lo que significa que resolverlo eficientemente implicaría resolver todos los problemas en $\mathcal{NP}$. Por otro lado, un problema es $\mathcal{NP}$ -hard si todo problema en $\mathcal{NP}$ puede reducirse a él en tiempo polinomial, aunque no necesariamente pertenezca a $\mathcal{NP}$. Esto incluye muchos problemas de optimización, cuya solución no solo es difícil de encontrar, sino también de verificar eficientemente.

Una gran parte de los problemas de optimización combinatoria relevantes en la práctica, como el *Travelling Salesman Problem* (TSP), el *Knapsack Problem* (KP) y el *Quadratic Assignment Problem* (QAP), son considerados $\mathcal{NP}$ -hard en su formulación de optimización. En general, sus versiones de decisión, que consisten en determinar si existe una solución cuyo costo no supere un umbral dado K , pertenecen a la clase de los problemas $\mathcal{NP}$ -completes [Garey y Johnson, 1979]. La dificultad computacional de estos problemas ha impulsado el desarrollo de métodos aproximados, entre los cuales destacan las heurísticas y metaheurísticas, capaces de encontrar soluciones de alta calidad en tiempos razonables.

Entre los problemas más representativos se incluyen el *Boolean Satisfiability Problem* (SAT), el *Travelling Salesman Problem* (TSP), el *Vehicle Routing Problem* (VRP), el *Knapsack Problem* (KP), y el *Quadratic Assignment Problem* (QAP), entre otros. Cada uno de ellos presenta estructuras con restricciones específicas y complejidades particulares que los hacen atractivos

para el estudio algorítmico. En muchos casos, estos problemas son $\mathcal{NP}$ -hard en su versión de optimización, y $\mathcal{NP}$ -complete en su versión de decisión.

Para los objetivos de esta memoria se trabajará con el *Travelling Salesman Problem* (TSP) y el *Quadratic Assignment Problem* (QAP), los cuales se presentan a continuación.

2.1.1. Travelling Salesman Problem (TSP)

El *Travelling Salesman Problem* (TSP), también conocido como *Travelling Salesperson Problem*, es uno de los problemas más conocidos y estudiados en el campo de la optimización combinatoria. Este problema inició su formulación informal en contextos de planificación desde las décadas de 1930 y 1940, y fue formalizado en las décadas siguientes, convirtiéndose en un problema central para el análisis de la complejidad algorítmica y los métodos de búsqueda en espacios de soluciones discretos [Lawler, 1985].

En términos generales, el TSP plantea el desafío de encontrar la ruta más corta que permita a un viajero recorrer un conjunto de n ciudades, visitando cada una exactamente una vez y regresando al punto de partida. El objetivo es minimizar el costo total del recorrido, ya sea en distancia, tiempo o algún otro criterio relevante.

Para representar este problema de forma matemática, se utiliza un grafo completo $G = (V, E)$, donde $V = \{v_1, v_2, \dots, v_n\}$ es el conjunto de vértices que representan las ciudades, y E es el conjunto de aristas que conectan cada par de ciudades. A cada arista $(v_i, v_j) \in E$ se le asigna un valor $d_{ij} \in \mathbb{R}^+$ que corresponde al costo de trasladarse entre v_i y v_j .

El recorrido buscado corresponde a un ciclo hamiltoniano⁵ de costo mínimo. Formalmente, se trata de encontrar una permutación $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ de los vértices que minimice la siguiente expresión:

$$\min_{\pi \in S_n} \sum_{i=1}^n d_{\pi_i, \pi_{i+1}}, \quad \text{donde } \pi_{n+1} := \pi_1, \quad (2)$$

donde S_n representa el conjunto de todas las permutaciones posibles de n elementos. Al definir π_{n+1} como el primer vértice del recorrido, se asegura que la suma incluya también el retorno a la ciudad de origen, cerrando el ciclo de forma completa.

2.1.2. Quadratic Assignment Problem (QAP)

El *Quadratic Assignment Problem* (QAP) es un problema clásico en optimización combinatoria, introducido formalmente por Koopmans y Beckmann en la década de 1950 para modelar problemas de ubicación de instalaciones [Koopmans y Beckmann, 1957]. Desde entonces,

⁵Secuencia cerrada de vértices en un grafo que visita exactamente una vez cada vértice y retorna al punto de inicio.

ha sido ampliamente estudiado debido a su aplicabilidad y a la complejidad computacional que presenta, siendo considerado uno de los problemas más difíciles dentro de la clase $\mathcal{NP}$ -hard.

En términos generales, el QAP plantea la tarea de asignar un conjunto de n instalaciones a n ubicaciones, de forma que se minimice el costo total asociado tanto a las distancias entre ubicaciones como a los flujos de interacción entre instalaciones. La dificultad del problema radica en que el costo no depende solamente de asignaciones individuales, sino de pares de ellas, lo que introduce una componente cuadrática en la función objetivo.

Para formalizar el problema, se requiere la definición de dos matrices: una matriz de flujo $F = [f_{ij}] \in \mathbb{R}^{n \times n}$, donde f_{ij} representa la cantidad de flujo entre las instalaciones i y j , y una matriz de distancias $D = [d_{kl}] \in \mathbb{R}^{n \times n}$, donde d_{kl} es la distancia entre las ubicaciones k y l . El objetivo es encontrar una permutación $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ que minimice el costo total:

$$\min_{\pi \in S_n} \sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{\pi_i \pi_j}, \quad (3)$$

donde S_n es el conjunto de todas las permutaciones posibles de n elementos. En esta formulación, π_i indica la ubicación asignada a la instalación i , y el término $f_{ij} \cdot d_{\pi_i \pi_j}$ representa el costo inducido por asignar la instalación i a la ubicación π_i y la instalación j a la ubicación π_j .

2.2. Técnicas Completas

Las técnicas completas son enfoques algorítmicos que garantizan encontrar una solución óptima a un problema de optimización, siempre que exista, y asegurar su inexistencia en caso contrario. Esto se logra mediante una exploración sistemática del espacio de soluciones, asegurando la cobertura total de este [Russell y Norvig, 2020].

Estas técnicas son especialmente útiles en dominios donde la precisión es crítica y es necesario garantizar soluciones óptimas, como ocurre en la verificación formal⁶, la programación entera o los problemas de satisfacción de restricciones (CSPs).

En general, este tipo de técnicas se apoya en un modelo estructurado del espacio de búsqueda (como un árbol o grafo), una estrategia sistemática de exploración (por ejemplo, búsqueda en profundidad o amplitud), y mecanismos de poda que permiten reducir el número de alternativas a considerar sin comprometer la completitud. Algunas técnicas relacionadas a estos ejemplos son algoritmos de *Backtracking* y *Branch and Bound* [Crama et al., 1995].

A pesar de sus ventajas, la aplicabilidad de estas técnicas se ve limitada por su alto costo

⁶Consiste en demostrar matemáticamente que un sistema cumple con una especificación lógica, usando modelos como autómatas, lógica temporal o teoría de conjuntos.

computacional. Problemas combinatorios como el TSP o el QAP tienen espacios de búsqueda que crecen factorial o exponencialmente respecto al tamaño del problema, haciendo que incluso instancias moderadas resulten intratables para una técnica completa.

2.3. Técnicas Incompletas

Las técnicas incompletas son enfoques de resolución que, a diferencia de los métodos completos, no garantizan encontrar la solución óptima ni asegurar su inexistencia. Su objetivo es obtener soluciones de buena calidad en tiempos razonables, especialmente cuando el espacio de búsqueda es demasiado amplio para explorarse exhaustivamente. Este tipo de estrategias comenzó a desarrollarse con mayor fuerza entre los años 1960 y 1970, impulsado por el crecimiento de la computación y la necesidad de enfrentar problemas combinatorios de alta complejidad [Glover, 1986].

En general, estas técnicas se basan en la exploración parcial del espacio de soluciones, guiada por mecanismos que priorizan regiones prometedoras. Su funcionamiento suele apoyarse en conceptos como el *vecindario* (*neighborhood*), definido como el conjunto de soluciones cercanas a otra según un operador específico de movimiento, y la *búsqueda local* (*local search*), que aplica iterativamente dichos operadores con el objetivo de mejorar la solución actual dentro del *vecindario* de esta misma. En este contexto, un operador consiste en una regla que modifica una solución candidata dada para generar otra dentro del mismo vecindario; por ejemplo, en el TSP, un operador clásico es el *2-opt*, que elimina dos aristas del recorrido actual y las reconecta de manera distinta para generar una nueva solución vecina [Crama *et al.*, 1995, Aarts y Lenstra, 1997].

Durante este proceso pueden alcanzarse soluciones que, aunque no sean globalmente óptimas, no pueden mejorarse por ninguno de sus vecinos inmediatos: estas se conocen como óptimos locales [Aarts y Lenstra, 1997]. Dado que las técnicas incompletas no exploran todo el espacio, es posible que queden atrapadas en estas soluciones. Por lo que para superar esta dificultad, muchas estrategias modernas incorporan mecanismos de intensificación (o explotación), que profundizan en zonas prometedoras, y diversificación (o exploración), que permiten escapar de regiones subóptimas y alcanzar nuevas áreas del espacio de búsqueda [Blum y Roli, 2003].

Las técnicas incompletas se dividen comúnmente en dos grandes familias: las heurísticas y las metaheurísticas. Las primeras están basadas en reglas específicas de decisión aplicadas directamente al problema, mientras que las segundas definen marcos de resolución más generales y adaptables a diferentes problemas.

2.3.1. Heurísticas

Las heurísticas son técnicas que utilizan reglas empíricas, intuiciones de diseño o conocimientos específicos sobre un problema para construir soluciones rápidamente. No requieren una exploración completa del espacio de búsqueda y, aunque no garantizan optimalidad, pueden producir soluciones aceptables con bajo esfuerzo computacional [Pearl, 1984].

El uso del término “heurística” en el contexto computacional fue impulsado en la década de 1950, en parte por los trabajos pioneros de Newell y Simon sobre la resolución de problemas humanos y la modelación del razonamiento inteligente mediante sistemas simbólicos [Simon y Newell, 1971]. Desde entonces, las heurísticas han sido ampliamente adoptadas como estrategias efectivas para abordar problemas complejos como el TSP, el VRP y el QAP, entre otros.

Entre las heurísticas más conocidas se encuentra el método del *Nearest Neighbor* (NN) [Rosenkrantz *et al.*, 1977], utilizado en problemas como el TSP, donde se construye una solución eligiendo en cada paso la opción más cercana disponible. El *Random Walk* [Papadimitriou, 1991], que consiste en una estrategia de exploración aleatoria, sin guía determinista, siendo útil en contextos donde se busca evitar sesgos o escapar de óptimos locales. Por otro lado, la heurística de *Greedy Construction* [Edmonds, 1971] consiste en construir una solución paso a paso, seleccionando en cada etapa la opción con el mayor beneficio inmediato, sin considerar su efecto global en la solución final. Este enfoque ha demostrado ser efectivo en diversos problemas combinatorios, como el KP, donde se priorizan elementos con alta relación de valor y peso en función de una métrica local.

2.3.2. Metaheurísticas

Las metaheurísticas son marcos algorítmicos con un propósito general diseñados para abordar problemas complejos de optimización que resultan difíciles de resolver mediante técnicas completas, ya sea por su elevada complejidad computacional o por la naturaleza no lineal, multimodal o discreta del espacio de búsqueda. Estas estrategias se apoyan en heurísticas de alto nivel que permiten explorar el espacio de soluciones de forma eficiente, buscando un equilibrio entre exploración e intensificación para alcanzar soluciones de alta calidad sin necesidad de un conocimiento detallado del problema. Una de las principales fortalezas de las metaheurísticas es su versatilidad y capacidad de adaptación a una amplia gama de problemas, independientemente del dominio de aplicación o de la estructura específica del problema. Esto contrasta con las heurísticas tradicionales, que suelen estar diseñadas para resolver un único problema o una clase muy limitada de problemas [Abdel-Basset *et al.*, 2018].

Estas técnicas han sido ampliamente aplicadas en planificación de la producción, bioinformática, logística, ingeniería, entre otros dominios. A lo largo del tiempo, se han desarrollado enfoques inspirados en fenómenos diversos, incluyendo procesos naturales (como la evolución biológica, el comportamiento colectivo de insectos o el movimiento de partícu-

las), mecanismos físicos (como el enfriamiento de metales), e incluso procesos cognitivos o sociales. Aunque existen clasificaciones tradicionales basadas en la forma de exploración —como los métodos que operan sobre una única solución (trayectoriales) frente a los que utilizan poblaciones—, el campo ha evolucionado hacia propuestas híbridas y adaptativas que combinan distintos principios para mejorar su eficacia.

En este contexto, a continuación se presentan algunas de las metaheurísticas más representativas, las cuales ejemplifican los distintos enfoques y principios que han guiado el desarrollo de estas técnicas.

Simulated Annealing (SA) [Kirkpatrick *et al.*, 1983] Es una metaheurística trayectorial estocástica inspirada en el proceso físico de recocido térmico, en el que un material fundido se enfría gradualmente para alcanzar un estado de mínima energía. Esta analogía fue propuesta por Kirkpatrick, Gelatt y Vecchi en 1983, al adaptar el *Metropolis algorithm* —una regla de aceptación probabilística utilizada para simular equilibrio térmico— al contexto de la optimización combinatoria. Dicha regla acepta siempre soluciones mejores, y peores con una probabilidad decreciente dependiente de la temperatura, lo que favorece la exploración del espacio de búsqueda.

SA inicia desde una solución inicial y la modifica iterativamente mediante la generación de nuevas soluciones en su *vecindario*, definido por un operador que determina las posibles transiciones. Su naturaleza estocástica permite aceptar soluciones peores con una probabilidad dada por $\exp(-\Delta E/T)$, donde ΔE es el incremento en la función objetivo y T representa la temperatura. Este mecanismo facilita escapar de óptimos locales y explorar regiones diversas del espacio. El Algoritmo 1 resume su funcionamiento para el caso de minimización, con los parámetros iniciales: solución x_0 , temperatura inicial T_0 , temperatura mínima $T_{\min}$, y factor de enfriamiento α con $0 < \alpha < 1$.

Diversas extensiones de SA han sido propuestas a lo largo del tiempo, modificando tanto el esquema de enfriamiento como los criterios de término. En el plano teórico, Geman y Geman [Geman y Geman, 1984] demostraron que un factor de enfriamiento logarítmico $T(t) \sim \frac{c}{\log(1+t)}$ garantiza la convergencia al óptimo global bajo ciertas condiciones de equilibrio y *ergodicidad*⁷. En la práctica, se han preferido esquemas geométricos con factor fijo α o enfoques adaptativos que ajustan la reducción de la temperatura según el avance de la búsqueda [Aarts y Korst, 1989]. Asimismo, se han explorado criterios de término distintos a una temperatura mínima, como la falta de mejoras tras cierto número de iteraciones o la estabilización del valor objetivo.

⁷En este contexto implica que el proceso estocástico tiene la capacidad de explorar todo el espacio de soluciones con el tiempo suficiente, asegurando que la probabilidad de alcanzar cualquier estado no nulo se mantiene en el largo plazo.

Algorithm 1 Simulated Annealing (SA) [Kirkpatrick *et al.*, 1983]

Require: $x_0, T_0, T_{\min}, \alpha$

```

1:  $x \leftarrow x_0$ 
2:  $T \leftarrow T_0$ 
3: while  $T > T_{\min}$  do
4:    $x' \leftarrow \text{GenerateNeighbor}(x)$ 
5:    $\Delta E \leftarrow f(x') - f(x)$ 
6:   if  $\Delta E < 0$  then
7:      $x \leftarrow x'$ 
8:   else
9:      $p \leftarrow \exp(-\Delta E/T)$ 
10:    if  $\text{Random}(0, 1) < p$  then
11:       $x \leftarrow x'$ 
12:    end if
13:  end if
14:   $T \leftarrow \alpha \cdot T$ 
15: end while
16: Output:  $x$ 

```

Tabu Search (TS) [Glover, 1990] Es una metaheurística de optimización basada en *búsqueda local*, diseñada para superar las limitaciones de los métodos clásicos, en particular su tendencia a quedar atrapados en óptimos locales. Fue propuesta por Fred Glover a fines de la década de 1980 como una estrategia sistemática para guiar la exploración del espacio de soluciones más allá de enfoques puramente *greedy*, mediante el uso de estructuras de memoria que registran información sobre el historial de la búsqueda.

A partir de una solución inicial, TS ejecuta iterativamente una *búsqueda local* generando un conjunto de soluciones vecinas, y selecciona aquella que presenta el mejor valor de evaluación, incluso si empeora la solución actual. Para evitar ciclos o recorridos repetitivos, mantiene una estructura llamada *lista tabú* (*tabu list*), la cual almacena atributos de los últimos movimientos realizados, prohibiendo su repetición durante un número limitado de iteraciones. Excepcionalmente, un movimiento catalogado como tabú puede aceptarse si cumple un criterio de aspiración, es decir, si permite acceder a una solución aún no visitada o mejora una condición predefinida. El procedimiento general de TS se presenta en el Algoritmo 2, donde se definen los parámetros iniciales: la solución inicial x_0 , el tamaño de la *lista tabú* t y el número máximo de iteraciones MaxIter .

Desde su formulación original, TS ha sido ampliada con componentes de memoria de mediano y largo plazo, mecanismos explícitos de intensificación y diversificación, y estrategias dinámicas para ajustar el tamaño de la *lista tabú*, lo que le otorga una notable capacidad de adaptación a diferentes tipos de problemas.

Algorithm 2 Tabu Search (TS) [Glover, 1990]

Require: $x_0, t, \text{MaxIter}$

```

1:  $x \leftarrow x_0$ 
2:  $x^* \leftarrow x$ 
3:  $T \leftarrow \emptyset$ 
4: for  $k = 1$  to  $\text{MaxIter}$  do
5:    $N \leftarrow \text{GenerateNeighbors}(x)$ 
6:    $N' \leftarrow \{x' \in N \mid \text{Move}(x, x') \notin T \text{ or satisfies aspiration}\}$ 
7:    $x' \leftarrow \arg \min_{x' \in N'} f(x')$ 
8:    $T \leftarrow \text{UpdateTabuList}(T, \text{Move}(x, x'), t)$ 
9:    $x \leftarrow x'$ 
10:  if  $f(x) < f(x^*)$  then
11:     $x^* \leftarrow x$ 
12:  end if
13: end for
14: Output:  $x^*$ 

```

Genetic Algorithms (GA) [Holland, 1992] Los GA constituyen una familia de metaheurísticas inspiradas en los principios de la evolución biológica, introducidas formalmente por John Holland en la década de 1970. Su funcionamiento se basa en la adaptación mediante selección natural y reproducción, donde una población de soluciones evoluciona a través de operadores que simulan procesos como el cruce y la mutación genética.

A diferencia de las técnicas basadas en *búsqueda local*, que trabajan sobre una única solución, los GA operan sobre una población de individuos que representa distintas soluciones candidatas, codificadas usualmente como cadenas binarias o vectores. Su proceso evolutivo se estructura en torno a tres operadores fundamentales:

- *selección (selection)*: determina qué individuos participarán en la etapa de reproducción, y es uno de los principales responsables de la presión selectiva impuesta por el algoritmo. Entre los métodos más comunes se encuentran el *roulette wheel selection*, donde cada individuo recibe una probabilidad proporcional a su aptitud; y el *tournament selection*, donde se elige al mejor dentro de un subconjunto aleatorio de tamaño k (conocido como *tournament size*). Este parámetro permite ajustar la presión selectiva: valores mayores de k favorecen la explotación, mientras que valores menores promueven la diversidad.
- *Cruce (crossover)*: combina la información genética de dos padres para generar nuevos descendientes. El método más clásico es el *one-point crossover*, que intercambia segmentos a partir de un punto aleatorio del cromosoma; otras variantes, como el *uniform crossover*, mezclan genes de forma distribuida. Este operador se aplica con una probabilidad p_c .
- *mutación (mutation)*: introduce alteraciones aleatorias en los genes de los descendien-

tes, con el fin de mantener la diversidad genética y evitar la convergencia prematura. Se aplica con una baja probabilidad p_m , que debe balancear la exploración sin afectar negativamente la estabilidad evolutiva.

El procedimiento general se resume en el Algoritmo 3, donde se destacan los parámetros fundamentales: tamaño de la población n , número de generaciones MaxGen, probabilidad de cruce p_c y probabilidad de mutación p_m .

Algorithm 3 Genetic Algorithms (GA) [Holland, 1992]

Require: $n, \text{MaxGen}, p_c, p_m$

```

1:  $P \leftarrow \text{InitializePopulation}(n)$ 
2: for  $g = 1$  to MaxGen do
3:    $P' \leftarrow \emptyset$ 
4:   while  $|P'| < n$  do
5:      $p_1, p_2 \leftarrow \text{SelectParents}(P)$ 
6:     if  $\text{Random}(0, 1) < p_c$  then
7:        $c_1, c_2 \leftarrow \text{Crossover}(p_1, p_2)$ 
8:     else
9:        $c_1, c_2 \leftarrow p_1, p_2$ 
10:    end if
11:     $c_1 \leftarrow \text{Mutate}(c_1, p_m)$ 
12:     $c_2 \leftarrow \text{Mutate}(c_2, p_m)$ 
13:     $P' \leftarrow P' \cup \{c_1, c_2\}$ 
14:  end while
15:   $P \leftarrow \text{EvaluateAndSelect}(P \cup P')$ 
16: end for
17: Output: Best(P)

```

Desde su formulación original, los GA han dado lugar a numerosas variantes y extensiones orientadas a mejorar su eficacia y adaptabilidad. Algunas propuestas se enfocan en modificar los criterios de selección, como los esquemas con múltiples etapas de *tournament selection* por generación, o en redefinir condiciones de término basadas en la convergencia de la población o en la pérdida de diversidad genética. Otras variantes introducen mecanismos adaptativos que ajustan dinámicamente los valores de p_c y p_m según el estado evolutivo del algoritmo, promoviendo un balance más efectivo entre exploración y explotación. También se han desarrollado enfoques híbridos, como los *Memetic Algorithms*, que incorporan métodos de *búsqueda local* dentro del ciclo evolutivo para refinar soluciones localmente. Por otro lado, las estructuras poblacionales distribuidas han dado lugar a modelos como los *island models*, donde la población global se divide en subpoblaciones que evolucionan de forma semi-independiente con migraciones ocasionales; y los *Parallel Genetic Algorithms*, que paralelizan la evolución para mejorar tanto la eficiencia computacional como la diversidad exploratoria [Ong et al., 2006, Cantu-Paz, 2000].

Particle Swarm Optimization (PSO) [Kennedy y Eberhart, 1995] Es una metaheurística diseñada para la optimización continua, la cual fue propuesta por Kennedy y Eberhart en 1995, inspirada en el comportamiento colectivo de organismos sociales como bandadas de aves o cardúmenes de peces. En este modelo, las partículas —que representan soluciones candidatas— ajustan su trayectoria combinando su experiencia individual con la influencia del grupo, lo que permite una exploración colaborativa del espacio de búsqueda.

Una representación visual de este comportamiento puede observarse en la Figura 2, donde se ejemplifica cómo aves o peces se organizan dinámicamente, combinando aprendizaje local con coordinación global.

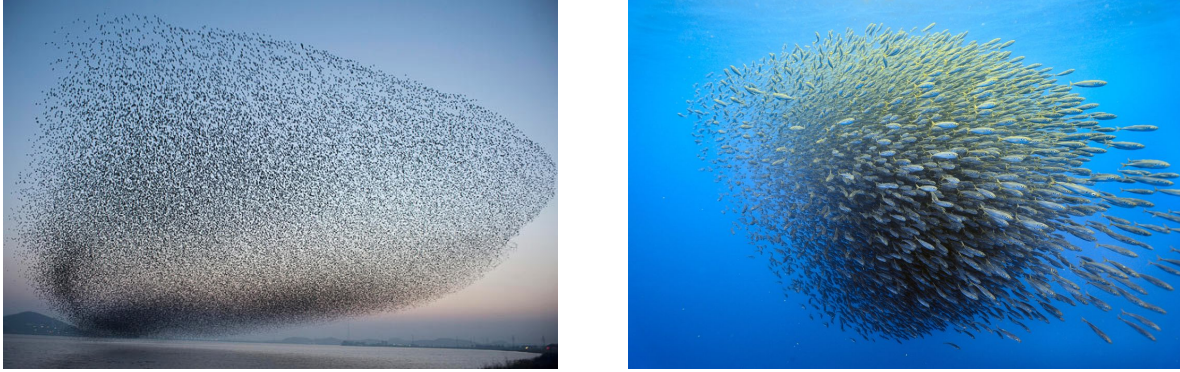


Figura 2: Representación del comportamiento colectivo observado en bandadas de aves y cardúmenes de peces.

Fuentes: [Ballardini, 2018, Daley, 2015]

Formalmente, cada partícula se caracteriza por una posición x_i y una velocidad v_i dentro de un espacio continuo. Su movimiento se actualiza iterativamente a partir de tres componentes: un término de inercia que conserva parte de su dirección anterior, una atracción hacia la mejor solución que ha alcanzado (*personal best*, $pbest_i$), y otra hacia la mejor solución global conocida por el enjambre (*global best*, $gbest$). Esta dinámica se modela con las siguientes ecuaciones:

$$v_i \leftarrow w \cdot v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i) \quad (4)$$

$$x_i \leftarrow x_i + v_i \quad (5)$$

donde w es el coeficiente de inercia, c_1 y c_2 ponderan las influencias individual y social respectivamente, y r_1, r_2 son valores aleatorios en $[0, 1]$.

El procedimiento general se resume en el Algoritmo 4. Inicialmente, las partículas se ubican aleatoriamente dentro del espacio de búsqueda, definido en n dimensiones acotadas por los límites $[b_{lo}, b_{up}]$, y se les asigna una velocidad inicial proporcional al rango del dominio. Durante la ejecución, cada partícula actualiza su estado conforme a su rendimiento y el del enjambre, repitiendo este proceso durante un número máximo de iteraciones (MaxIter).

PSO requiere como parámetros iniciales: el número de partículas S , la dimensión del espacio de búsqueda n , los límites $[b_{lo}, b_{up}]$ para cada dimensión, y el número máximo de iteraciones

MaxIter. Asimismo, deben definirse inicialmente los coeficientes w , ϕ_p y ϕ_g , que controlan respectivamente la inercia del movimiento, la atracción hacia la mejor posición personal y la atracción hacia la mejor posición global conocida por el enjambre.

Algorithm 4 Particle Swarm Optimization (PSO) [Kennedy y Eberhart, 1995]

Require: S , n , $[b_{lo}, b_{up}]$, MaxIter, w , ϕ_p y ϕ_g

```

1: for each particle  $i = 1$  to  $S$  do
2:   Initialize  $x_i \sim \mathcal{U}(b_{lo}, b_{up})$ 
3:   Initialize  $pbest_i \leftarrow x_i$ 
4:   Evaluate  $f(pbest_i)$ 
5:   if  $f(pbest_i) < f(gbest)$  then
6:      $gbest \leftarrow pbest_i$ 
7:   end if
8:   Initialize  $v_i \sim \mathcal{U}(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$ 
9: end for
10: for  $t = 1$  to MaxIter do
11:   for each particle  $i = 1$  to  $S$  do
12:     for each dimension  $d = 1$  to  $n$  do
13:        $r_1, r_2 \sim \mathcal{U}(0, 1)$ 
14:        $v_{i,d} \leftarrow w \cdot v_{i,d} + \phi_p \cdot r_1 \cdot (pbest_{i,d} - x_{i,d}) + \phi_g \cdot r_2 \cdot (gbest_d - x_{i,d})$ 
15:        $x_{i,d} \leftarrow x_{i,d} + v_{i,d}$ 
16:     end for
17:     Evaluate  $f(x_i)$ 
18:     if  $f(x_i) < f(pbest_i)$  then
19:        $pbest_i \leftarrow x_i$ 
20:       if  $f(pbest_i) < f(gbest)$  then
21:          $gbest \leftarrow pbest_i$ 
22:       end if
23:     end if
24:   end for
25: end for
26: Output:  $gbest$ 

```

Desde su introducción, PSO ha dado lugar a numerosas variantes orientadas a mejorar el equilibrio entre exploración y explotación. Entre ellas destacan las topologías de comunicación alternativas, como el modelo *lbest*, donde cada partícula considera solo un subconjunto de vecinas para actualizar su trayectoria. También se han propuesto esquemas adaptativos que ajustan dinámicamente los parámetros w , c_1 y c_2 en función del progreso de la búsqueda. Además, se han incorporado mecanismos como el *velocity clamping*, que limita la magnitud de las velocidades, y el uso de un *constriction coefficient* para garantizar la estabilidad del movimiento [Clerc y Kennedy, 2002]. Otras extensiones incluyen versiones híbridas con *búsqueda local*, y enfoques como *multi-swarm PSO*, donde distintos enjambres interactúan entre sí para preservar la diversidad y evitar la convergencia prematura [Shi y Eberhart, 1998, Blackwell y Branke, 2004].

Ant Colony Optimization (ACO) [Dorigo et al., 1996] Es una metaheurística bioinspirada propuesta por Dorigo et al., basada en el comportamiento colectivo de las hormigas al buscar caminos cortos entre su nido y una fuente de alimento. Este comportamiento emerge de una forma de comunicación indirecta conocida como *estigmergia*, donde los individuos coordinan sus acciones modificando el entorno —en este caso, mediante la deposición de feromonas sobre los trayectos recorridos— sin necesidad de comunicación directa. La probabilidad de que una hormiga elija un determinado camino depende tanto de la intensidad de feromonas acumuladas como de una heurística local que representa la calidad del trayecto.

El mecanismo de decisión se modela mediante una regla de transición probabilística, donde la probabilidad de que la hormiga k visite a su próximo nodo j desde el nodo actual i , es:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} & \text{si } j \in N_i^k \\ 0 & \text{en otro caso} \end{cases} \quad (6)$$

donde $\tau_{ij}(t)$ es la cantidad de feromonas sobre el arco (i, j) en el tiempo t , η_{ij} es el valor heurístico (por ejemplo, $1/d_{ij}$ en el TSP), N_i^k es el conjunto de nodos no visitados por la hormiga k , y los parámetros α y β controlan la influencia relativa de la feromona y de la heurística, respectivamente.

Una vez construidas las soluciones, las feromonas se actualizan considerando tanto su evaporación como los depósitos realizados por las hormigas, de acuerdo con:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (7)$$

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q}{L_k} & \text{si la hormiga } k \text{ utilizó el arco } (i, j) \\ 0 & \text{en otro caso} \end{cases} \quad (8)$$

donde ρ es el coeficiente de evaporación, Q es una constante que determina la cantidad de feromonas depositadas, y L_k es el costo de la solución generada por la hormiga k .

El procedimiento general de ACO se resume en el Algoritmo 5, el cual contempla una fase inicial de configuración, seguida por ciclos iterativos de construcción de soluciones y actualización de feromonas. Para su ejecución, ACO requiere definir como parámetros iniciales: el número de hormigas m , los coeficientes α y β que controlan la influencia relativa de la feromona y la heurística, el coeficiente de evaporación ρ , la constante de depósito Q , y el número máximo de iteraciones MaxIter .

Algorithm 5 Ant Colony Optimization (ACO) [Dorigo *et al.*, 1996, Pei *et al.*, 2012]

Require: $m, \alpha, \beta, \rho, Q, MaxIter$

- 1: Initialize pheromones $\tau_{ij}(0) = c$ for all edges (i, j)
 - 2: **for** $t = 1$ **to** $MaxIter$ **do**
 - 3: **for** each ant $k = 1$ **to** m **do**
 - 4: Construct a valid solution using the probabilistic rule (Eq. 1)
 - 5: Evaluate the solution and compute L_k
 - 6: **end for**
 - 7: **for** each edge (i, j) **do**
 - 8: Compute $\Delta\tau_{ij}^k$ using Eq. 3-4
 - 9: Update $\tau_{ij}(t + 1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \sum_k \Delta\tau_{ij}^k$
 - 10: **end for**
 - 11: Record the best global or local solution
 - 12: **end for**
 - 13: **Output:** Best solution found
-

Desde su formulación original como *Ant System*, el enfoque ha evolucionado hacia diversas variantes que buscan mejorar su eficiencia, estabilidad y capacidad de convergencia. *Ant Colony System* (ACS) [Dorigo y Gambardella, 1997] introduce mecanismos adicionales para intensificar la búsqueda, como una regla pseudo-aleatoria de decisión que alterna entre elección determinista y probabilística, y una actualización local de feromonas que fomenta una exploración más refinada del entorno de soluciones. Posteriormente, *MAX-MIN Ant System* (MMAS) [Stützle y Hoos, 1998] incorpora límites explícitos para los valores de feromona y restringe su actualización únicamente a la mejor hormiga, ya sea global o de la iteración, lo que reduce la volatilidad del proceso. Luego, *Rank-based Ant System* [Bullnheimer *et al.*, 1999] propone un esquema de ponderación que asigna mayor influencia a las mejores soluciones según su orden relativo. Finalmente, *Elitist Ant System* [White *et al.*, 2003] refuerza con feromonas adicionales los mejores caminos encontrados, acelerando la convergencia hacia soluciones prometedoras.

2.3.3. Dependencia de parámetros en técnicas incompletas

Independientemente de su enfoque, tanto las heurísticas como las metaheurísticas requieren definir valores para un conjunto de parámetros que afectan directamente su comportamiento y desempeño. Estos parámetros pueden controlar desde aspectos estructurales del algoritmo hasta detalles críticos de ejecución. Esta dependencia introduce un nuevo nivel de complejidad al problema de optimización original, ya que la calidad de los resultados puede variar significativamente según la configuración utilizada.

Para abordar adecuadamente esta problemática, es fundamental entender la naturaleza de los parámetros involucrados, los cuales se suelen clasificar en tres cate-

rías [López-Ibáñez *et al.*, 2016]:

- Cateóricos: Parámetros con valores discretos sin un orden implícito. Por ejemplo, el operador de selección utilizado en un GA, como *tournament selection* o *roulette wheel selection*.
- Ordinales: Parámetros con valores discretos que poseen una relación de orden natural. Por ejemplo, el tamaño de la *lista tabú* en TS.
- Numéricos: Parámetros con dominios continuos o enteros. Por ejemplo, la probabilidad de mutación en un GA, expresada como un valor real entre 0 y 1.

Adicionalmente, existen parámetros subordinados, cuya relevancia depende del valor asignado a otros parámetros. Por ejemplo, en un GA que implemente múltiples operadores de selección, el parámetro *tournament size* solo es relevante si se ha seleccionado el método de *tournament selection*; en caso contrario, modificar su valor no tiene ningún efecto sobre el comportamiento del algoritmo.

Aunque todas las técnicas están sujetas a esta dependencia, las metaheurísticas son particularmente sensibles al ajuste de parámetros. A diferencia de muchas heurísticas, que incorporan directamente conocimiento del problema para guiar su comportamiento, las metaheurísticas operan como marcos generales y requieren una adecuada selección de parámetros para adaptarse eficazmente a la estructura del problema específico. Esta necesidad se ve acentuada por la diversidad de tipos de parámetros involucrados y por las complejas interacciones entre ellos, que dificultan su ajuste manual. En consecuencia, surge el conocido *Parameter Setting Problem*, un desafío que consiste en identificar las configuraciones adecuadas para un algoritmo determinado en un conjunto de instancias de un problema dado, lo cual exige un enfoque sistemático y un esfuerzo significativo para lograr un rendimiento eficaz en distintas instancias del problema [Montero *et al.*, 2014].

2.4. Parameter Setting Problem

El Parameter Setting Problem es un desafío ampliamente estudiado en la literatura, ya que la eficacia de los algoritmos metaheurísticos depende en gran medida de la adecuada selección de los valores de sus parámetros [Montero *et al.*, 2014]. Este problema consiste en identificar las configuraciones que optimicen el rendimiento del algoritmo, ya sea en términos de calidad de la mejor solución encontrada, tiempo necesario para obtener una solución u otros criterios específicos. Sin embargo, determinar dichas configuraciones no es trivial, puesto que su evaluación requiere múltiples ejecuciones del algoritmo. En algoritmos estocásticos, es necesario evaluar cada configuración con distintas semillas para estimar su desempeño de forma robusta, lo que incrementa significativamente el costo computacional del ajuste. A

ello se suma que las configuraciones adecuadas pueden variar significativamente entre diferentes problemas o incluso entre distintas instancias del mismo problema, lo que introduce una alta dependencia contextual. Además, los parámetros suelen tener efectos interdependientes y no lineales sobre el comportamiento del algoritmo, lo que complica aún más su ajuste manual y dificulta una evaluación consistente.

Para abordar este problema, se han desarrollado el control de parámetros [Hinterding *et al.*, 1999] y la configuración automática de parámetros [Hutter, 2009].

2.4.1. Control de parámetros

El control de parámetros consiste en modificar los valores de los parámetros de un algoritmo metaheurístico de forma dinámica durante su ejecución. Este introduce mecanismos internos en el algoritmo que permiten ajustar estos valores en tiempo de ejecución, favoreciendo la adaptabilidad a distintas fases del proceso de búsqueda. Esta estrategia busca mejorar el equilibrio entre exploración y explotación a medida que evoluciona el estado del sistema, lo cual es especialmente beneficioso en contextos donde el comportamiento óptimo del algoritmo varía con el tiempo o depende de características emergentes del problema.

Las técnicas de control de parámetros se dividen típicamente en tres subcategorías [Hinterding *et al.*, 1999, Karafotias *et al.*, 2015]:

- **Determinístico:** Los valores de los parámetros varían de acuerdo con una regla explícita y predefinida que depende del número de iteraciones, del tiempo transcurrido o de alguna función matemática. Estas estrategias no incorporan retroalimentación del estado del algoritmo, y su simplicidad las hace apropiadas para entornos controlados o bien estudiados.
- **Adaptativo:** El ajuste de los parámetros se realiza en función de información recolectada durante la ejecución, como la tasa de éxito de operadores, la diversidad de la población, o la calidad de las soluciones generadas. Se utilizan señales de desempeño para modificar los parámetros de forma reactiva, permitiendo una respuesta flexible ante cambios en el entorno o el progreso de la búsqueda.
- **Auto-adaptativo:** Los parámetros son codificados dentro de los individuos del algoritmo y evolucionan conjuntamente con las soluciones. En este esquema, el proceso evolutivo actúa también sobre los parámetros, seleccionando y recombinando configuraciones que resultan efectivas. Este enfoque es común en algoritmos evolutivos como Evolution Strategies (ES) y Differential Evolution (DE), donde la adaptación está embebida directamente en la representación de las soluciones.

El control de parámetros ha demostrado ser particularmente efectivo en problemas dinámi-

cos, en optimización con ruido (*noisy*⁸), y en escenarios donde el comportamiento óptimo del algoritmo varía a lo largo del tiempo o entre regiones del espacio de búsqueda. Al permitir una adaptación continua y contextualizada, este enfoque contribuye a mejorar la eficiencia, la robustez y la capacidad de generalización de los algoritmos metaheurísticos. Por ejemplo, en [Doerr, 2019], se estudia el control dinámico de la probabilidad de mutación en algoritmos evolutivos aplicados a funciones de referencia, mostrando mejoras significativas frente a configuraciones estáticas.

A pesar de sus ventajas, el control de parámetros presenta desafíos importantes. Entre ellos destacan la dificultad de diseñar mecanismos de ajuste que resulten efectivos sin incurrir en una sobrecarga computacional significativa, añadir nuevos parámetros en la definición, la definición de reglas lo suficientemente generales, y la necesidad de definir métricas de desempeño confiables en enfoques adaptativos. En el caso del control auto-adaptativo, la evolución de parámetros puede conducir a configuraciones ineficaces si no se diseñan adecuadamente los espacios de representación y operadores. Además, la naturaleza dinámica de estos métodos complica su análisis teórico y validación empírica. Estos desafíos han sido ampliamente discutidos en la literatura, donde también se señala su sensibilidad a hiperparámetros y aplicabilidad limitada según el algoritmo o dominio [Gomes Pereira de Lacerda *et al.*, 2021].

2.4.2. Configuración automática de parámetros

La configuración automática de parámetros, también conocida como sintonización de parámetros, es una estrategia que busca determinar, mediante procedimientos sistemáticos y automatizados, la mejor combinación de valores para los parámetros de un algoritmo antes de su ejecución. Este proceso se basa en la evaluación iterativa de múltiples configuraciones sobre un conjunto de instancias de entrenamiento (idealmente representativas del dominio), con el objetivo de identificar aquella que optimice una función de desempeño específica (usualmente asociada a calidad de solución, tiempo de ejecución, robustez, o una combinación de estos).

Formalmente, el problema de configuración de algoritmos puede modelarse como la 7-tupla $\langle \Theta, I, p_I, p_C, t, C, T \rangle$ [Birattari *et al.*, 2010], donde:

- Θ es el conjunto de configuraciones candidatas del algoritmo objetivo.
- I representa el conjunto de instancias del problema.
- p_I es una medida de probabilidad definida sobre I , que permite modelar la distribución de las instancias.
- $p_C(c \mid \theta, i)$ es la medida de probabilidad que modela la distribución del coste c al ejecutar la configuración $\theta \in \Theta$ sobre la instancia $i \in I$.

⁸Consiste a la variabilidad aleatoria o incertidumbre en las observaciones, evaluaciones o mediciones, que impide obtener resultados exactos o deterministas en determinados datos.

- $t : I \rightarrow \mathbb{R}^+$ es una función que asigna a cada instancia i un tiempo máximo de cómputo permitido.
- $C \subseteq \mathbb{R}$ denota el dominio de los posibles valores de coste.
- T representa el presupuesto total disponible (por ejemplo, en tiempo o número de evaluaciones) para realizar el proceso de sintonización.

El objetivo consiste en encontrar una configuración $\bar{\theta} \in \Theta$ que minimice el valor esperado del coste, considerando tanto la variabilidad entre instancias como la estocasticidad del algoritmo:

$$\bar{\theta} \in \arg \min_{\theta \in \Theta} C(\theta) \quad (9)$$

donde $C(\theta)$ representa el valor esperado del coste inducido por la configuración θ , definido como:

$$C(\theta) = \mathbb{E}_{I,C}[c] = \int c \cdot dp_C(c \mid \theta, i) \cdot dp_I(i) \quad (10)$$

Esta formulación permite capturar tanto la diversidad de instancias del problema como la variabilidad inherente de los algoritmos estocásticos. Además, puede adaptarse a otros criterios de evaluación, tales como la mediana, percentiles o métricas multiobjetivo, según las necesidades del contexto de aplicación.

Sin embargo, esta generalidad y flexibilidad conllevan un costo computacional considerable. Cada evaluación de una configuración requiere la ejecución efectiva del algoritmo sobre una o más instancias, lo que implica un consumo elevado de recursos. Esta situación se agrava aún más en el caso de algoritmos estocásticos, donde es necesario realizar múltiples repeticiones con diferentes semillas aleatorias para obtener una estimación robusta de su desempeño.

A pesar de estos desafíos, la sintonización de parámetros ha cobrado especial relevancia en la última década, dado su impacto positivo en la mejora del rendimiento algorítmico. En la literatura se encuentran numerosos casos de aplicación exitosa, donde su uso ha permitido optimizar significativamente el comportamiento de algoritmos complejos en contextos de optimización combinatoria, planificación y aprendizaje automático. Por ejemplo, *irace* ha sido ampliamente utilizado en la calibración de algoritmos evolutivos para problemas clásicos como el *Travelling Salesman Problem* y variantes del *Vehicle Routing Problem* [López-Ibáñez *et al.*, 2016]. Asimismo, *ParamILS* ha mostrado mejoras sustanciales al configurar automáticamente algoritmos de resolución del Boolean Satisfiability Problem (SAT), como *SPEAR*, reduciendo considerablemente los tiempos de ejecución frente a configuraciones predeterminadas [Hutter *et al.*, 2009].

Estas aplicaciones han motivado el desarrollo de diversas estrategias y herramientas específicas para enfrentar el problema de configuración automática. A continuación, se presentan algunas de las más relevantes, tanto por su impacto práctico como por su adopción en distintos dominios:

- ParamILS [Hutter *et al.*, 2009]: Emplea una estrategia de *iterated local search* para explorar el espacio de configuraciones. Su versión FocusedILS permite comparar configuraciones con diferentes niveles de evaluación, maximizando la eficiencia de la búsqueda.
- F-RACE [Birattari *et al.*, 2010]: Basado en técnicas estadísticas de *racing*, evalúa configuraciones progresivamente y elimina las menos prometedoras a través de pruebas de hipótesis (como el test de Friedman⁹).
- irace [López-Ibáñez *et al.*, 2016]: Generaliza F-RACE incorporando una estrategia adaptativa de generación de configuraciones candidatas, ajustando el proceso de *muestreo* en función del rendimiento observado.
- EVOCA [Riff y Montero, 2013]: Diseñado sobre un esquema evolutivo, explora el espacio de configuraciones mediante operadores genéticos. Está especialmente orientado a espacios de búsqueda complejos, incluyendo parámetros subordinados o con fuertes dependencias condicionales.
- GGA++ [Ansótegui *et al.*, 2015]: Variante de un algoritmo genético híbrido para la configuración automática de algoritmos GRASP. Introduce mecanismos específicos de codificación y selección que favorecen la adaptación a distintos dominios y restricciones propias del problema de configuración.
- SMAC [Hutter *et al.*, 2011]: Emplea modelos predictivos, como *random forests*, para estimar el rendimiento de configuraciones no evaluadas. Su enfoque secuencial equilibra de manera efectiva la exploración de nuevas configuraciones con la explotación de aquellas que han mostrado buen desempeño, siendo especialmente adecuado para espacios con parámetros mixtos y condicionales.

Entre todos ellos, *irace* ha destacado por su versatilidad y resultados consistentes en una amplia gama de problemas. Por ello, será considerado como el sintonizador en este estudio.

2.5. *irace*

irace es un sintonizador de parámetros que combina evaluación secuencial, selección estadística y generación adaptativa de configuraciones. Su diseño está inspirado en técnicas de *racing*, donde su objetivo es concentrar el presupuesto computacional en aquellas configuraciones con mayor evidencia de buen desempeño, descartando progresivamente las menos prometedoras mediante pruebas estadísticas. Esta lógica permite explorar eficientemente grandes espacios de parámetros sin evaluar exhaustivamente todas las combinaciones [López-Ibáñez *et al.*, 2016].

⁹Prueba estadística no paramétrica usada para comparar múltiples configuraciones sobre varias instancias.

El funcionamiento de *irace* toma como base los principios de *F-Race* e *Iterated F-Race*, algoritmos que introdujeron la evaluación progresiva de configuraciones mediante pruebas estadísticas como el test no paramétrico de Friedman, una prueba utilizada para comparar múltiples configuraciones sobre varias instancias sin asumir normalidad en los datos. En lugar de evaluar todas las configuraciones en cada instancia, estas técnicas permiten descartar rápidamente aquellas que muestran un rendimiento inferior de forma consistente. Para superar las limitaciones de escalabilidad de estas primeras versiones, Balaprakash et al. propusieron el algoritmo *Iterated F-Race (I/F-Race)* [Balaprakash et al., 2007], el cual incorporó mecanismos de *muestreo* probabilístico y un enfoque iterativo que permite reutilizar el conocimiento de configuraciones exitosas para guiar nuevas generaciones de configuraciones, lo que derivó en el desarrollo del *irace* package con capacidades mejoradas de adaptación y control.

A partir de estas ideas, *irace* construye un ciclo de mejora iterativa guiado por información estadística acumulada. La Figura 3 ilustra esquemáticamente este proceso de sintonización, destacando las fases de evaluación comparativa, selección de élites y generación de nuevas configuraciones mediante modelos adaptativos de *muestreo*.

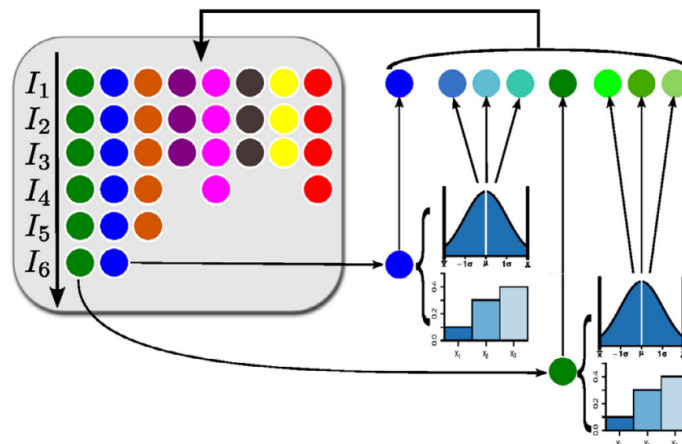


Figura 3: Esquema general del funcionamiento de *irace*.

Fuente: [López-Ibáñez et al., 2016].

A la izquierda de la figura se observa el proceso de evaluación mediante *racing*, donde configuraciones candidatas (representadas por colores distintos) son comparadas sobre un conjunto creciente de instancias. Las configuraciones con peor desempeño son descartadas con base en pruebas estadísticas, como el test de Friedman (por ejemplo, las configuraciones café, amarillo y morado son descartadas tras la ejecución de la instancia 3). Solo las configuraciones con mejor rendimiento (las sobrevivientes) forman el conjunto élite.

Este conjunto élite alimenta el ajuste de los modelos de *muestreo*, que estiman distribuciones de probabilidad sobre cada parámetro a partir de los valores observados. Estos modelos, ilustrados mediante curvas y diagramas de frecuencia, concentran la probabilidad en regiones del espacio que han mostrado buen desempeño. A partir de ellos, se generan nuevas

configuraciones (representadas a la derecha), que participan en la siguiente iteración junto a las élites anteriores, manteniendo un balance entre explotación de soluciones conocidas y exploración de alternativas prometedoras. Este proceso continúa hasta agotar el presupuesto definido.

El procedimiento general de *irace*, previamente ilustrado en la Figura 3, puede formalizarse mediante el Algoritmo 6. En la fase inicial se genera un conjunto de configuraciones mediante *muestreo* probabilístico uniforme sobre el espacio de parámetros definido por el usuario. Este conjunto es evaluado a través de una etapa de *racing*, en la cual se eliminan progresivamente las configuraciones menos competitivas mediante las pruebas estadísticas, y se selecciona un subconjunto compuesto por aquellas configuraciones con mejor rendimiento relativo.

A partir de estas élites iniciales comienza un ciclo iterativo, en el que se generan nuevas configuraciones guiadas por modelos de *muestreo* ajustados con base en los parámetros observados en las configuraciones élites previas. Estas nuevas configuraciones se combinan con la élite anterior para formar un conjunto ampliado que es evaluado nuevamente mediante *racing*. Las mejores configuraciones de esta evaluación conforman la nueva élite, que se utilizará en la siguiente iteración. Este proceso continúa mientras quede presupuesto disponible.

Algorithm 6 *irace* [López-Ibáñez *et al.*, 2016]

Require: $X, I, C(\theta, i), T$

```

1:  $T_{used} \leftarrow 0$ 
2:  $\Theta_1 \leftarrow \text{SampleUniform}(X)$ 
3:  $I_1 \subseteq I$ 
4:  $\Theta^{elite} \leftarrow \text{Race}(\Theta_1, I_1, T_{used})$ 
5:  $j \leftarrow 1$ 
6: while  $T_{used} < T$  do
7:    $j \leftarrow j + 1$ 
8:    $\Theta^{new} \leftarrow \text{Sample}(X, \Theta^{elite})$ 
9:    $\Theta_j \leftarrow \Theta^{new} \cup \Theta^{elite}$ 
10:   $I_j \subseteq I$ 
11:   $\Theta^{elite} \leftarrow \text{Race}(\Theta_j, I_j, T_{used})$ 
12: end while
13: return  $\Theta^{elite}$ 

```

En esta formulación, se definen las siguientes variables:

- Θ_j representa el conjunto de configuraciones candidatas generadas en la iteración j .
- Θ^{elite} es el subconjunto de Θ_j compuesto por las configuraciones con mejor rendimiento esperado; se utiliza como base para actualizar el modelo de *muestreo*.

- X define el espacio paramétrico sobre el cual se construyen las configuraciones $\theta \in \Theta$, incluyendo dominios, restricciones y condiciones de activación.
- $I_j \subseteq I$ es el subconjunto de instancias utilizado durante la iteración j .
- $C(\theta, i)$ es la función de coste que evalúa el desempeño de la configuración θ sobre la instancia $i \in I$.

El comportamiento de `irace` está determinado no solo por su estructura iterativa, sino también por un conjunto de componentes internos y parámetros de configuración que controlan aspectos críticos del proceso de búsqueda. Entre los parámetros más relevantes se encuentra el número de instancias necesarias antes de aplicar el primer test estadístico, que condiciona cuán pronto se pueden descartar configuraciones de bajo rendimiento; así como el tamaño mínimo del conjunto élite, que controla el grado de presión en la selección sobre las configuraciones supervivientes. Valores bajos de este parámetro promueven una convergencia rápida pero arriesgan una exploración limitada, mientras que tamaños mayores permiten mantener diversidad a costa de una mayor cantidad de evaluaciones. Otro hiperparámetro fundamental es el nivel de significancia estadística α utilizado en los tests de descarte (típicamente 0.05), cuyo ajuste determina el grado de exigencia al momento de eliminar configuraciones. Además, `irace` permite elegir entre distintos tests estadísticos (como t-test¹⁰ o Friedman), definir la política de selección de instancias (aleatoria o secuencial), y especificar cómo manejar los *outliers*¹¹ en las métricas de coste.

Dentro de este conjunto de componentes también se encuentra el mecanismo de *soft-restart*, que actúa como estrategia de recuperación frente a la convergencia prematura. Este mecanismo detecta cuándo las configuraciones generadas en una iteración son excesivamente similares entre sí, lo cual puede indicar que el modelo de *muestreo* se ha concentrado demasiado en una región del espacio de búsqueda. En tal caso, el modelo se reinicia parcialmente para restaurar la diversidad, reutilizando versiones anteriores o ampliando las distribuciones del *muestreo*.

Todos estos hiperparámetros pueden ser ajustados por el usuario y tienen un impacto directo sobre la eficacia del proceso de configuración. Este impacto ha sido analizado en profundidad por Pérez Cáceres et al. [Pérez Cáceres *et al.*, 2014], quienes evaluaron empíricamente cómo distintas configuraciones afectan el rendimiento global de `irace` en tres escenarios distintos: ACOTSP, SPEAR y MOACO. El estudio consideró cinco hiperparámetros clave: el número de iteraciones, la cantidad mínima de instancias antes del primer test estadístico (`firstTest`), el tamaño máximo del conjunto élite, el tipo de test de descarte y su nivel de significancia α .

Los resultados muestran que el valor de `firstTest` influye directamente en la agresividad del descarte inicial, donde valores bajos permiten una eliminación temprana de configuracio-

¹⁰Prueba estadística paramétrica utilizada para comparar medias entre dos grupos relacionados o dependientes.

¹¹Datos que se desvían significativamente del resto y que pueden distorsionar análisis estadísticos si no son manejados adecuadamente.

nes no prometedoras, pero aumentan el riesgo de descartar prematuramente configuraciones potencialmente buenas. En cuanto al tamaño del conjunto élite, se observó que valores pequeños favorecen una rápida convergencia, pero pueden limitar la exploración del espacio de búsqueda; mientras que tamaños mayores promueven diversidad, a costa de un mayor número de evaluaciones. Finalmente, el estudio también identificó que la elección del test estadístico (t-test versus Friedman) y el nivel de significancia α afectan de manera distinta dependiendo del escenario, siendo el t-test más robusto en presencia de *noisy* u *outliers*.

Estos hallazgos refuerzan la importancia de comprender el efecto de cada hiperparámetro y de ajustar sus valores en función del problema y del algoritmo objetivo, ya que una configuración informada puede reducir considerablemente el número de evaluaciones necesarias para alcanzar configuraciones de alto rendimiento. Sin embargo, más allá del ajuste individual de estos parámetros, es igualmente relevante analizar cómo se organiza la búsqueda en el espacio de configuraciones a lo largo del tiempo. Comprender el comportamiento exploratorio que subyace al proceso de *irace* permite no solo mejorar su eficiencia, sino también revelar patrones o limitaciones inherentes a su dinámica de *muestreo*. En este contexto, las *Search Trajectory Networks* se presentan como una herramienta poderosa para estudiar las trayectorias recorridas durante la sintonización. Este enfoque permite visualizar el recorrido exploratorio, identificar zonas de estancamiento, regiones de alta concentración o sesgos del modelo de *muestreo*, así como detectar configuraciones particularmente influyentes en el proceso.

2.6. Search Trajectory Networks

Search Trajectory Networks (STNs) son un tipo de modelo basado en grafos dirigidos propuesto para representar de manera explícita las trayectorias recorridas por algoritmos estocásticos durante el proceso de búsqueda [Ochoa *et al.*, 2021]. Su estructura permite capturar la secuencia completa de soluciones exploradas por el algoritmo a lo largo del tiempo, reflejando su comportamiento dinámico dentro del espacio de búsqueda. Esta representación facilita el análisis estructural del proceso exploratorio, permitiendo identificar patrones de movimiento, zonas de alta concentración, retrocesos y trayectorias recurrentes. La Figura 4 muestra un ejemplo esquemático de una STN, en la cual cada nodo representa una región del espacio visitada por el algoritmo. El tamaño, la forma y el color de los nodos codifican información relevante sobre su rol y características (por ejemplo, frecuencia de visita o posición dentro de la trayectoria). Los arcos dirigidos indican las transiciones observadas entre regiones consecutivas, conservando el orden temporal, y su color refleja la calidad asociada a dicha transición.

Este enfoque se inspira en los modelos conocidos como *Local Optima Networks* (LONs) [Ochoa *et al.*, 2014], los cuales representan los *fitness landscapes*¹² mediante gra-

¹²Representación del espacio de soluciones donde a cada solución se le asigna un valor que indica su calidad o aptitud.

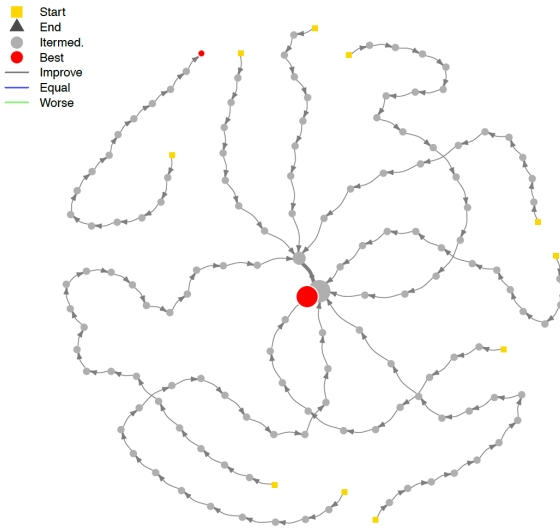


Figura 4: Ejemplo de una STN.
Fuente: [Ochoa, 2021]

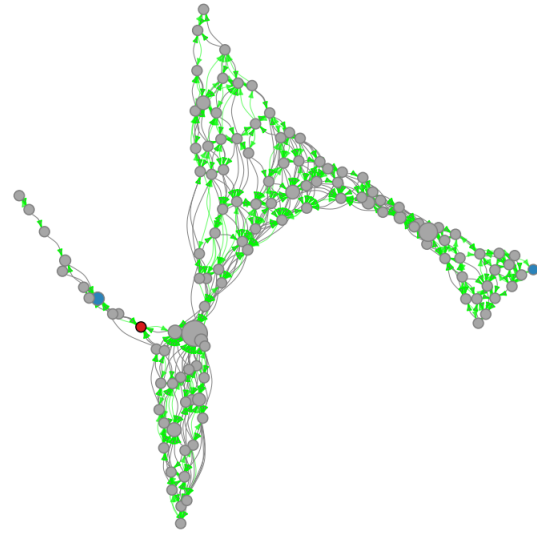


Figura 5: Ejemplo de una LON.
Fuente: [Cleghorn y Ochoa, 2021]

fos dirigidos, donde los nodos corresponden a óptimos locales y los arcos indican posibles transiciones entre ellos. La Figura 5 muestra un ejemplo de LON en el que el color, tamaño y forma de los nodos contiene información sobre su rol (óptimo regular, *sink* óptimo¹³ o no óptimo). Los arcos reflejan la calidad de la transición, los cuales son: verde (mejora), negro (empeoramiento) y azul (neutral). Esta representación permite identificar propiedades estructurales del *landscape*, como *big valleys*¹⁴ y *funnels*¹⁵.

No obstante, al modelar únicamente los óptimos locales, las LONs descartan la secuencia de soluciones que el algoritmo recorre para llegar a dichas regiones. Las STNs superan esta restricción al incorporar todo el recorrido del algoritmo, capturando no solo óptimos locales, sino también las soluciones visitadas que conforman las trayectorias del proceso del algoritmo. Esto permite obtener una visión más detallada del comportamiento del algoritmo, incorporando tanto los caminos exitosos como los intentos fallidos o desviaciones, lo cual resulta fundamental para entender su funcionamiento en profundidad [Treimun-Costa *et al.*, 2020, Cleghorn y Ochoa, 2021].

2.6.1. Definición del modelo

Una STN se construye a partir de la trayectoria generada por un algoritmo estocástico durante su ejecución. Para modelar esta información de forma estructurada, es necesario realizar una serie de definiciones fundamentales del modelo [Ochoa *et al.*, 2021]:

¹³Óptimo local sin arcos salientes que puede representar el óptimo global.

¹⁴Estructura del *landscape* donde los óptimos locales tienden a agruparse cerca del óptimo global.

¹⁵Regiones del *landscape* donde múltiples trayectorias convergen hacia un mismo óptimo.

- **Solución representativa:** Es la solución seleccionada en un determinado momento del proceso de búsqueda para representar el estado del algoritmo. Su elección depende del tipo de estrategia utilizada por el algoritmo, por ejemplo, en algoritmos basados en poblaciones, se suele utilizar a la mejor solución de la población en un determinado momento.
- **Locación:** Corresponde a una región del espacio de búsqueda definida, a partir de una partición previa respecto al espacio de soluciones del algoritmo, donde cada solución pertenece exclusivamente a una locación. A cada locación se le asigna un valor representativo de calidad, calculado según un criterio específico, como el valor mínimo, el promedio o alguna otra medida relevante para el análisis.
- **Search Trajectory:** Es la secuencia ordenada de soluciones representativas registradas durante la ejecución del algoritmo. Estas soluciones se transforman en una secuencia de locaciones, lo que permite construir una trayectoria estructurada en el grafo.
- **Nodo:** Elemento del grafo que representa una locación identificada en alguna trayectoria. Los nodos pueden incluir atributos adicionales, como la cantidad de veces que fueron visitados o métricas agregadas de calidad, las cuales se suelen asociar a su forma o color.
- **Arco:** Conexión dirigida entre dos locaciones visitadas de manera consecutiva dentro de una trayectoria de búsqueda. Cada arco puede estar ponderado según la frecuencia con la que se ha observado dicha transición durante la construcción del modelo, y su codificación por color puede reflejar información adicional, como la variación de calidad entre ambas locaciones.

Con estas definiciones, una STN se define formalmente como un grafo dirigido $G = (V, E)$, donde:

- V es el conjunto de nodos, donde cada uno correspondiente a una locación visitada durante la trayectoria de búsqueda.
- $E \subseteq V \times V$ es el conjunto de arcos dirigidos que representan las transiciones observadas entre locaciones consecutivas en dicha trayectoria.

Cada ejecución independiente del algoritmo objetivo sobre una instancia da origen a una trayectoria de búsqueda, la cual representa una secuencia ordenada de soluciones exploradas. A partir de una o más de estas ejecuciones, se construye una STN, que consolida las transiciones observadas entre locaciones en un grafo dirigido. En este modelo, cada trayectoria individual se representa visualmente desde un nodo inicial (cuadrado) hasta un nodo final (triángulo), como se ilustra en la Figura 4, donde se presentan un total de 10 ejecuciones.

Además del modelo clásico, las STNs cuentan con una extensión denominada merged STN, que permite comparar el comportamiento de distintos algoritmos (o variantes de uno mismo) ejecutados sobre una misma instancia. Esta fusión se construye a partir de los modelos STN individuales, siempre que compartan la misma definición de locaciones y se refieran a una instancia común del problema.

La Figura 6 muestra un ejemplo de merged STN, construida a partir de las trayectorias de tres algoritmos distintos ejecutados sobre una misma instancia. Al igual que en las STNs convencionales, cada trayectoria se representa mediante un conjunto de nodos y arcos codificados por color. En este caso, cada color identifica a un algoritmo específico, mientras que los nodos grises indican locaciones compartidas entre dos o más trayectorias. El nodo rojo representa la mejor solución encontrada, independientemente del algoritmo que la haya alcanzado. Asimismo, los nodos de inicio y fin de cada trayectoria se mantienen representados mediante cuadrados amarillos y triángulos negros, respectivamente. Los arcos conservan el color del algoritmo al que pertenecen, lo que permite analizar visualmente los puntos de convergencia y divergencia entre las trayectorias exploradas por los distintos métodos.

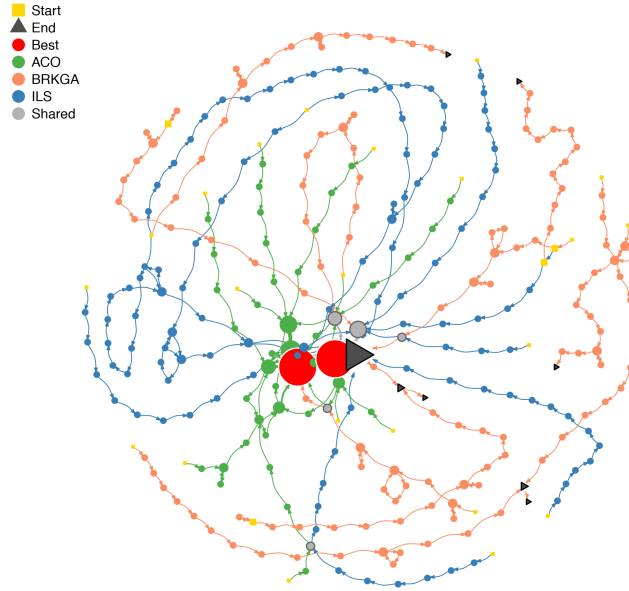


Figura 6: Ejemplo de merged STN.
Fuente: [Ochoa, 2021].

Formalmente, dada una instancia fija para un problema y dos algoritmos A y B , con sus respectivas STNs $G_A = (V_A, E_A)$ y $G_B = (V_B, E_B)$, se define la merged STN como el grafo dirigido:

$$G^* = (V^*, E^*) = (V_A \cup V_B, E_A \cup E_B) \quad (11)$$

El grafo G^* de la fusión de los grafos G_A y G_B contiene todos los nodos y arcos presentes en al menos uno de los modelos originales, donde los atributos asociados a estos permiten identificar a qué algoritmo pertenece cada elemento, facilitando la comparación visual y estructural de las trayectorias exploradas por cada enfoque.

Gracias a esta representación consolidada, el modelo resulta especialmente útil para analizar puntos de superposición o divergencia en las regiones exploradas, contrastar dinámicas de búsqueda entre algoritmos, y detectar patrones diferenciales de explotación o exploración dentro de un mismo espacio de solución.

2.6.2. Métricas

Las STNs, además de su valor visual como herramienta exploratoria, permiten aplicar métricas estructurales y dinámicas que cuantifican distintos aspectos del proceso de búsqueda. Estas métricas han sido utilizadas en diversos contextos y adaptadas tanto a STNs individuales como a merged STNs [Ochoa *et al.*, 2021, Narvaez-Teran *et al.*, 2021, Lavinas *et al.*, 2022, Riveros *et al.*, 2024, Hu *et al.*, 2023, Sarti *et al.*, 2022].

A continuación, se resumen las métricas más relevantes para STNs y merged STNs:

- *nodes*: Número total de nodos. Corresponde a la cantidad de locaciones distintas visitadas; mide la diversidad espacial de la búsqueda.
- *edges*: Número total de arcos. Indica la cantidad de transiciones únicas observadas entre locaciones consecutivas; una red más densa puede reflejar mayor dispersión.
- *n_best*: Número de nodos asociados a las mejores soluciones encontradas. Evalúa la cobertura del algoritmo sobre regiones de alta calidad.
- *n_end*: Número de nodos donde terminan las trayectorias. Útil para identificar zonas de convergencia o estancamiento.
- *n_shared*: En merged STNs, cantidad de nodos visitados por dos o más trayectorias provenientes de distintos algoritmos. Indica superposición en regiones exploradas.
- *mean_in, mean_out*: Grado promedio de entrada y salida. Miden la conectividad media de los nodos y la complejidad del flujo de la búsqueda.
- *max_in, max_out*: Grado máximo de entrada y salida. Permiten identificar nodos estructuralmente dominantes o críticos en la red.
- *best_strength*: Grado de entrada ponderado sobre los nodos de mejor calidad. Se interpreta como una medida de centralidad de las soluciones más prometedoras.
- *p_length*: Longitud promedio de los caminos más cortos desde los nodos iniciales hasta los mejores nodos. Mide la eficiencia estructural del proceso de búsqueda.
- *n_paths*: Número total de trayectorias con menor longitud hacia los nodos de mejor calidad. Refleja la accesibilidad de las buenas regiones del espacio.

- n_{comp} : Número de componentes conexas en la red. Permite analizar la fragmentación del modelo y detectar regiones exploradas de forma independiente.
- w_{edges} : Número de arcos asociados a transiciones con empeoramiento de calidad. Se interpreta como una métrica de comportamiento no-greedy o de exploración divergente.

2.6.3. Aplicaciones de STNs en la literatura

Las *Search Trajectory Networks* (STNs) han demostrado ser herramientas efectivas para caracterizar el comportamiento de algoritmos metaheurísticos. En el trabajo [Narvaez-Teran *et al.*, 2021], se aplicaron por primera vez al *Cyclic Bandwidth Sum Problem* (CBSP), un problema NP-hard cuyas soluciones se modelan como permutaciones. Para construir las STNs, se utilizó una técnica de reducción de dimensionalidad mediante *multidimensional scaling* (MDS), seguida de un agrupamiento tridimensional que permitió definir locaciones representativas del espacio de búsqueda.

Esta estructura fue utilizada para comparar el comportamiento de un *memetic algorithm* (MA) con una variante extendida basada en selección adaptativa, denominada *Dynamic Multi-Armed Bandit with Memetic Algorithm* (DMAB+MA). Las STNs construidas a partir de múltiples ejecuciones permitieron representar nodos como locaciones visitadas y arcos dirigidos como transiciones entre ellas. El análisis reveló que, mientras el MA producía trayectorias más dispersas y con mayor redundancia, el DMAB+MA generaba redes más compactas y dirigidas hacia regiones de alta calidad. Tal como se muestra en la Figura 7, la merged STN para una de las instancias evidencia estas diferencias en la organización y cobertura del espacio de búsqueda.

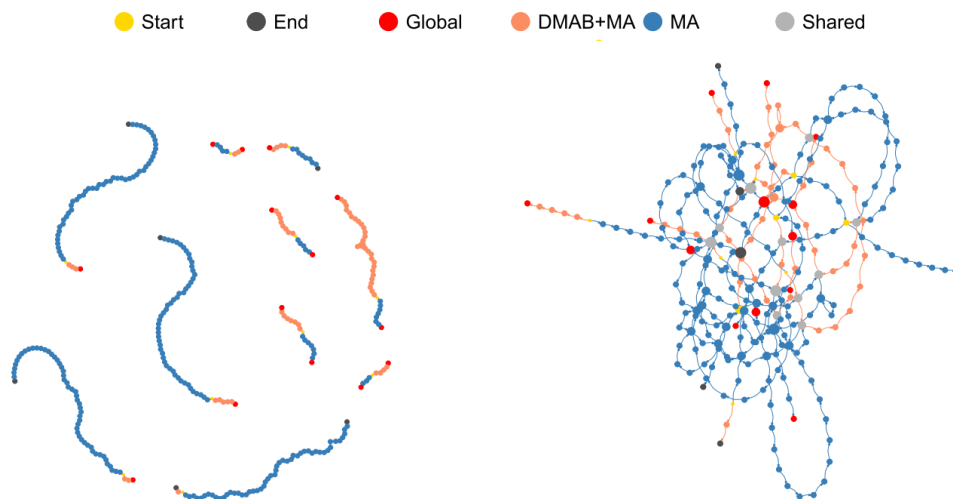


Figura 7: Merged STN construida para la instancia $p9c9$ del CBSP, combinando trayectorias del MA clásico y el DMAB+MA. Fuente: [Narvaez-Teran *et al.*, 2021].

Además del análisis visual, se emplearon métricas estructurales para caracterizar cuantitativamente el comportamiento exploratorio de los algoritmos. Se utilizaron las métricas *nodes* y *edges* para estimar la diversidad espacial y la complejidad de las trayectorias, observándose una mayor dispersión en el MA frente a rutas más dirigidas en DMAB+MA. Las métricas *n_best*, *n_end*, *p_length* y *n_paths* reflejaron que DMAB+MA alcanzaba con mayor frecuencia regiones de alta calidad mediante trayectorias más cortas y consistentes. Por el contrario, el MA presentó mayor fragmentación del espacio y transiciones de peor calidad, como lo evidencian las métricas *n_comp* y *w_edges*, lo que sugiere una exploración más errática y menos selectiva. Finalmente, en algunas instancias se observó que los nodos terminales no coincidían con los nodos óptimos, lo cual indica convergencia hacia regiones subóptimas.

En el contexto de la optimización multiobjetivo, Lavinas et al. [Lavinas et al., 2022] extendieron el modelo de STNs para analizar el comportamiento de algoritmos evolutivos como NSGA-II y MOEA/D-DE. Su enfoque se basó en la descomposición del problema en múltiples *scalar subproblems*, cada uno asociado a un vector de pesos. Para cada uno de estos vectores se extrajeron trayectorias de soluciones representativas, las cuales fueron discretizadas mediante una grilla de resolución 10^{-3} en el espacio objetivo, definiendo así las locaciones. A partir de estas, se construyeron STNs individuales por subproblema, que luego se fusionaron en una *merged STN* por cada algoritmo.

Esta estrategia permitió una comparación visual y estructural del comportamiento exploratorio de NSGA-II y MOEA/D. Tal como se muestra en la Figura 8, para el problema UF8, MOEA/D genera una red más conectada y distribuida, con soluciones óptimas alcanzadas tempranamente. En contraste, NSGA-II presenta una exploración más tardía, fragmentada y sin cobertura completa del frente de Pareto¹⁶.

Respecto al análisis estructural, las métricas *nodes* y *edges* permitieron estimar la diversidad espacial y complejidad del recorrido, reflejando que MOEA/D visitó más regiones únicas con trayectorias más densas. La métrica *n_shared* mostró menor redundancia en MOEA/D, mientras que *n_comp* evidenció mayor fragmentación en NSGA-II. El grado promedio y máximo (*mean_in*, *mean_out*, *max_in*, *max_out*) indicó una conectividad superior en MOEA/D, favoreciendo una exploración más cohesionada. Estas diferencias fueron consistentes con los resultados de calidad medidos mediante *hypervolume* e *IGD*, donde MOEA/D también mostró mejor rendimiento en términos de convergencia y diversidad.

En un estudio reciente, Riveros et al. [Riveros et al., 2024] extendieron el modelo de STNs al análisis de procesos de sintonización automática de parámetros, aplicándolo a dos enfoques: *ParamILS*, basado en *iterated local search*, y *Evoca*, un sintonizador de tipo poblacional. Ambos fueron empleados para ajustar los parámetros de un GA diseñado para resolver instancias del problema *NK-Landscape* [Kauffman, 1989], el cual modela funciones de aptitud altamente multimodales cuya complejidad aumenta con el parámetro de epistasis K , que regula la interdependencia entre variables.

¹⁶Conjunto de soluciones no dominadas en un problema multiobjetivo; representa el compromiso óptimo entre objetivos en conflicto.

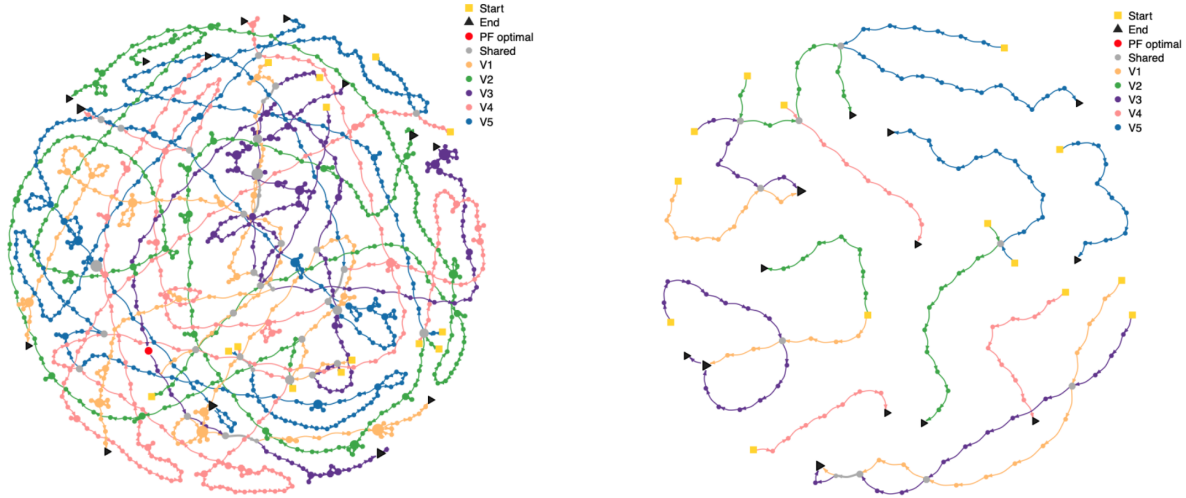


Figura 8: Merged STNs para el problema UF8. En la izquierda el algoritmo MOEA/D y a la derecha el NSGA-II respectivamente.

Fuente: [Lavinas *et al.*, 2022].

Para representar las trayectorias de búsqueda generadas por los sintonizadores, se construyeron STNs a partir de las configuraciones visitadas durante el proceso de ajuste. En el caso de ParamILS, las transiciones se derivaron del mecanismo de perturbación y mejora local propio de su estrategia, mientras que en Evoca se capturaron las mejores configuraciones generadas en cada generación de su dinámica evolutiva. Las locaciones fueron definidas mediante una discretización del espacio de configuraciones bajo dos niveles de granularidad (p_1 y p_2), lo que permitió generar redes con distinto nivel de resolución.

El análisis visual de las STNs reveló patrones estructurales contrastantes entre ambos sintonizadores. Tal como se observa en la Figura 9, correspondiente a la instancia 4-20 con precisión p_2 , la red generada por ParamILS muestra una estructura densa y fuertemente conectada, con trayectorias que convergen repetidamente en regiones similares. En cambio, la red de Evoca exhibe una estructura más fragmentada, con trayectorias dispersas y menor redundancia. Estas diferencias visuales reflejan el enfoque intensivo y local de ParamILS frente a la exploración amplia y diversa de Evoca.

Estas observaciones fueron respaldadas por métricas estructurales que cuantificaron las características topológicas de las redes. ParamILS produjo mayor número de nodos y arcos, múltiples trayectorias convergentes, caminos más largos y una mayor fuerza entrante hacia las configuraciones óptimas, lo que indica una exploración intensiva y centrada en zonas específicas. En contraste, Evoca presentó mayor número de componentes conexas, menor solapamiento, y mayor cobertura del espacio de búsqueda. Además, demostró una mayor robustez ante cambios experimentales: al aumentar el número de semillas para calcular la calidad de una configuración ($s = 30$), ParamILS redujo notablemente su conectividad, mientras que Evoca mantuvo una estructura coherente y estable.

Un hallazgo destacado fue el bajo grado de solapamiento entre las regiones exploradas por ambos métodos. En las *merged STNs*, construidas a partir de las trayectorias combinadas de ParamILS y Evoca sobre una misma instancia, se observó que solo un 3 % de los nodos fueron compartidos, lo que sugiere que los sintonizadores exploran regiones complementarias del espacio de configuraciones.

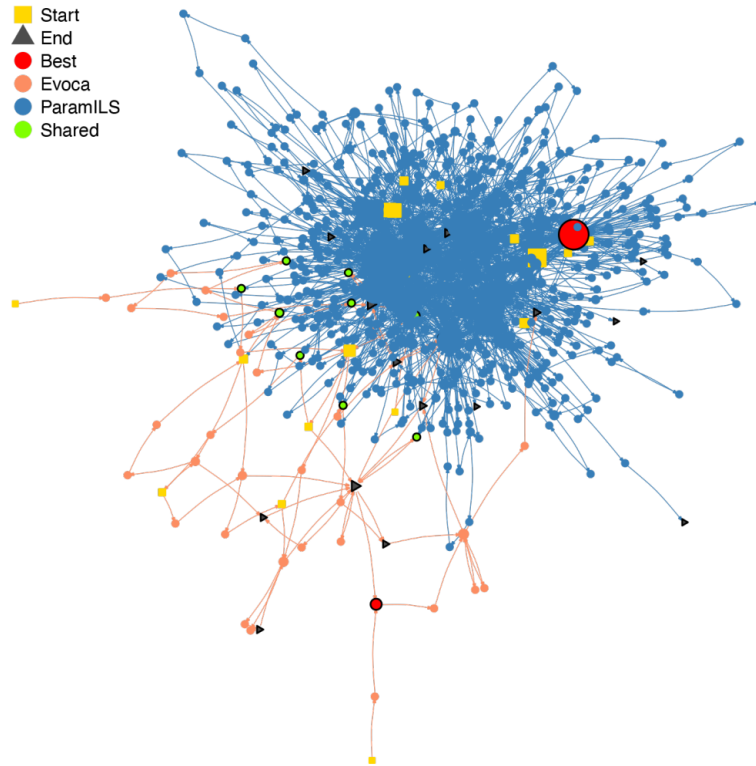


Figura 9: Merged STN para la instancia 4-20 con precisión p_2 y $s = 10$, combinando trayectorias de ParamILS y Evoca.

Fuente: [Riveros *et al.*, 2024].

Este estudio demuestra el potencial de las STNs para caracterizar de manera precisa y visualmente intuitiva las dinámicas de exploración y explotación en procesos de sintonización. Las métricas estructurales permiten identificar patrones complejos más allá del rendimiento final, lo que abre oportunidades para diseñar estrategias híbridas o adaptativas que aprovechen las fortalezas de distintos enfoques de ajuste automático de parámetros.

En este contexto, el análisis mediante STNs puede extenderse a otros sintonizadores como *irace*, permitiendo estudiar cómo se organiza su proceso de exploración, qué regiones del espacio de configuraciones prioriza y cómo actúa la presión selectiva impuesta por su modelo de *muestreo*. Esta perspectiva estructural no solo facilita el diagnóstico del comportamiento del algoritmo, sino que también ofrece una base cuantitativa para su ajuste, comparación con otros métodos y desarrollo de variantes más eficientes.

2.7. Resumen

En este capítulo se presentan los fundamentos clave sobre problemas de optimización, metaheurísticas y la dependencia de sus parámetros. Se definen formalmente problemas NP-hard como el *Travelling Salesman Problem* (TSP) y el *Quadratic Assignment Problem* (QAP), ampliamente utilizados en la literatura como escenarios de referencia para evaluar algoritmos de optimización. A continuación, se caracterizan cinco metaheurísticas representativas: *Simulated Annealing*, *Tabu Search*, *Genetic Algorithms*, *Particle Swarm Optimization* y *Ant Colony Optimization*, detallando su funcionamiento, componentes y operadores fundamentales. Se discute el rol crítico que cumplen sus parámetros y se clasifica su naturaleza en numéricos, categóricos, ordinales y subordinados, lo que da lugar al conocido *Parameter Setting Problem*.

Posteriormente, se detallan las estrategias de *parameter control* y *automated parameter tuning*, con especial énfasis en *irace*, el algoritmo de sintonización central en esta investigación. Se describe su funcionamiento, los mecanismos de *racing* iterativo, sus parámetros clave y el algoritmo general que emplea para explorar el espacio de configuraciones. Finalmente, se introduce el modelo de *Search Trajectory Networks* (STNs), una herramienta estructural para visualizar y cuantificar la dinámica de búsqueda de estos métodos. Se describen sus componentes, métricas analíticas y aplicaciones previas en contextos combinatorios, multiobjetivo y de configuración de parámetros, lo que proporciona el sustento teórico y metodológico para el diseño experimental y el análisis estructural del comportamiento exploratorio de *irace* en esta investigación.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

Este capítulo presenta la propuesta metodológica desarrollada para abordar el problema de investigación formulado en el Capítulo 1, cuyo objetivo es analizar estructuralmente el comportamiento exploratorio del sintonizador *irace* mediante el uso de modelos de *Search Trajectory Networks* (STNs). No obstante, el proceso de búsqueda llevado a cabo por *irace* no se ajusta directamente al concepto tradicional de trayectoria secuencial representado en los modelos STN clásicos, ya que emplea una estrategia basada en poblaciones iterativas y selección elitista. Esta característica imposibilita una aplicación directa del enfoque original, motivando la necesidad de adaptar dicho modelo a las particularidades del proceso de sintonización por configuraciones múltiples.

La solución se articula en torno a dos componentes principales. En primer lugar, se introduce el modelo *STN-i*, una adaptación del enfoque clásico de STNs al contexto de *irace*. Este modelo permite representar las trayectorias de búsqueda generadas durante la sintonización, incorporando elementos clave como la selección iterativa de configuraciones élite y las relaciones de descendencia entre configuraciones. Se definen además métricas estructurales y analíticas orientadas a caracterizar dichas trayectorias tanto de forma individual como comparativa, mediante la extensión denominada *merged STN-i*.

El segundo componente es el algoritmo *irace-STN-i*, encargado de transformar los resultados de ejecuciones independientes de *irace* en representaciones estructuradas compatibles con el modelo STN-i. Este proceso se organiza en cuatro etapas: definición de datos de entrada, lectura de datos, construcción de trayectorias y generación del archivo de salida.

3.1. Definición del modelo propuesto de STN adaptado a *irace* (STN-i)

Con el propósito de modelar de manera estructurada el comportamiento exploratorio del sintonizador *irace*, se propone el modelo *Search Trajectory Network* adaptado, denominado STN-i. Este modelo extiende el enfoque clásico de STNs para capturar la dinámica particular del proceso iterativo de *irace*, en el cual las configuraciones son generadas, evaluadas y descartadas de forma progresiva mediante pruebas estadísticas. En este contexto, las configuraciones élite —aquellas que superan el proceso de eliminación— cumplen un rol fundamental, ya que guían la construcción de nuevas configuraciones en iteraciones sucesivas. El modelo STN-i busca representar esta estructura de dependencia y selección, reflejando no solo qué configuraciones fueron evaluadas, sino también cómo se conectan y derivan unas de otras a lo largo del proceso de sintonización.

Formalmente, el modelo STN-i, al igual que el modelo STN clásico, se define como un grafo dirigido $G = (V, E)$, donde V es el conjunto de nodos que representan locaciones del es-

pacio de configuración, y E es el conjunto de arcos que capturan transiciones observadas entre dichas locaciones. En el caso de STN-i, estas transiciones son inducidas por el mecanismo iterativo y selectivo de *irace*, en el que las configuraciones son generadas, evaluadas y progresivamente descartadas mediante pruebas estadísticas. Este proceso da lugar a una dinámica de búsqueda estructurada en torno a las configuraciones seleccionadas, lo que se refleja en la distinción de dos tipos principales de nodos:

- *regular nodes*: representan locaciones que agrupan configuraciones descartadas en la misma iteración en que fueron generadas.
- *elite nodes*: representan locaciones que contienen al menos una configuración élite, es decir, aquellas seleccionadas al final de una iteración para ser utilizadas como base en la generación de nuevas configuraciones.

Al igual que en el modelo clásico de STNs, en STN-i se conserva la idea de clasificar los nodos según su rol topológico dentro de las trayectorias de búsqueda. No obstante, esta clasificación se adapta a las particularidades del funcionamiento iterativo de *irace*, considerando explícitamente la etapa en la que fueron generadas las configuraciones contenidas en cada nodo. Para ello, se utilizan tres categorías: START, STANDARD y END, definidas del siguiente modo:

- START: locaciones que contienen al menos una configuración generada en la primera iteración de *irace* (independiente de si es élite o no). Si además contiene alguna configuración final, se clasifica como END.
- END: locaciones que contienen al menos una configuración correspondiente a la última iteración de una ejecución particular de *irace* (es decir, la última configuración generada en su respectivo *run*).
- STANDARD: locaciones que no contienen configuraciones obtenidas en la primera o última iteración de una ejecución independiente de *irace*. Corresponden a regiones intermedias exploradas durante el proceso de búsqueda.

Dado que solo las configuraciones élite son utilizadas como base para la generación de nuevas configuraciones, las transiciones en el grafo son dirigidas exclusivamente por estas. Así, los nodos élite desempeñan un papel estructural fundamental al articular el avance exploratorio sobre el espacio de configuración. Las relaciones de influencia se modelan mediante el siguiente tipo de arco:

- *parent-edges*: arcos dirigidos desde un nodo élite hacia otro nodo, que corresponde a una locación generada en la iteración siguiente en el proceso iterativo de *irace*, representando la relación directa de influencia entre una configuración élite y sus configuraciones hijas. Estos arcos pueden conectar nodos élite con nodos regulares o élite.

De esta forma, el modelo STN-i permite, por tanto, reconstruir la trayectoria de búsqueda seguida por irace en términos estructurales, destacando las configuraciones clave, los puntos de bifurcación y las regiones de mayor persistencia. Esta representación facilita el análisis de aspectos como la convergencia, la diversidad exploratoria o la reutilización de regiones del espacio en distintas ejecuciones. Además, el modelo posibilita el análisis comparativo de múltiples ejecuciones independientes de irace bajo un mismo escenario de sintonización, es decir, aquellas que comparten el mismo algoritmo objetivo, conjunto de instancias y dominios de los parámetros. En este contexto, las locaciones actúan como discretizaciones del espacio continuo de configuración, permitiendo agrupar configuraciones similares y estudiar patrones comunes de exploración, convergencia o persistencia en diferentes ejecuciones.

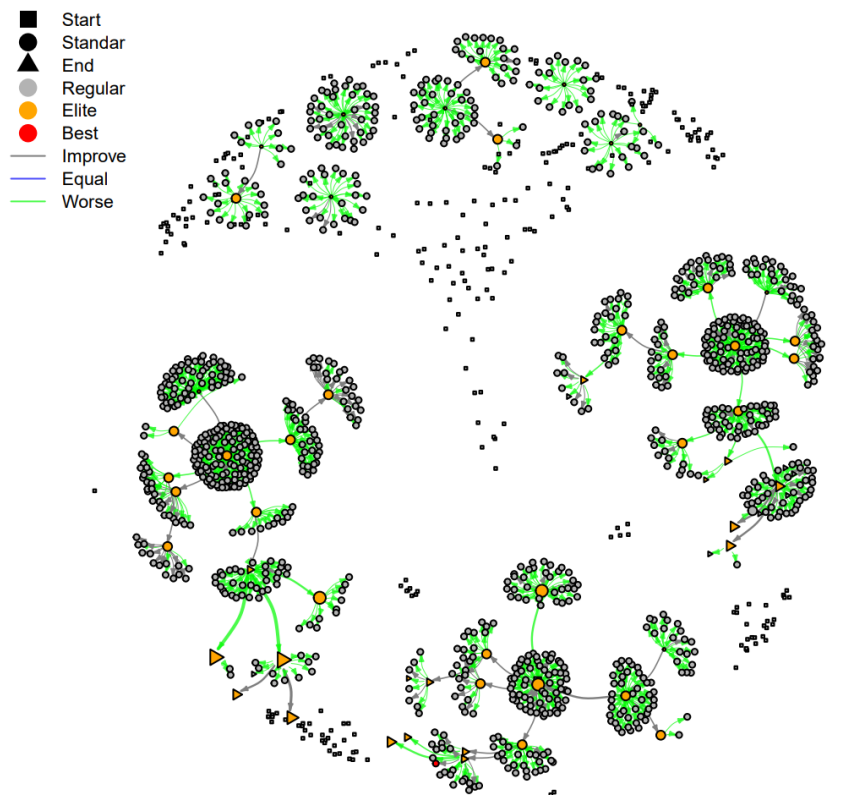


Figura 10: Ejemplo de STN-i generada a partir de múltiples ejecuciones independientes de irace.

Fuente: Elaboración propia.

La Figura 10 ilustra una instancia del modelo STN-i aplicado a múltiples ejecuciones independientes de irace. Los nodos se diferencian por estado (élite o regular) mediante el color y por su rol topológico (START, STANDARD, END) mediante su forma, mientras que los arcos representan transiciones inducidas exclusivamente por configuraciones élite.

Se observa cómo los nodos élite articulan el avance de la búsqueda, al ser los únicos que generan nuevas configuraciones, concentrando la mayoría de los arcos salientes. También es común la presencia de nodos START desconectados, que agrupan configuraciones iniciales

descartadas rápidamente, reflejando la capacidad de diversificación en el espacio de configuración.

El modelo también permite identificar casos en los que una o más configuraciones élite, agrupadas en una misma locación, pueden actuar como finales —representadas por triángulos amarillos— y, al mismo tiempo, generar descendencia al ser seleccionadas nuevamente, permaneciendo en la misma locación.

3.1.1. Merged STN-i

Siguiendo el principio del modelo clásico de merged STNs, se propone una variante adaptada a irace denominada merged STN-i. Esta extensión permite realizar análisis comparativos estructurados entre dos o más redes STN-i generadas a partir de ejecuciones independientes de irace, siempre que dichas redes compartan una definición común de locación (es decir, un mismo criterio de discretización del espacio de configuración), el mismo algoritmo objetivo, el conjunto de parámetros sintonizados, sus respectivos dominios y el conjunto de instancias de evaluación. Bajo estas condiciones, la merged STN-i facilita la comparación directa de regiones del espacio explorado, trayectorias de búsqueda o patrones de convergencia, en relación con diferencias controladas en los experimentos. Por ejemplo, es posible comparar dos redes STN-i construidas sobre una misma locación L_1 , donde una fue generada con el mecanismo de *soft-restart* habilitado y la otra sin dicho mecanismo, lo que permite estudiar el impacto de *soft-restart* en la diversidad de las configuraciones exploradas por el sintonizador.

En este modelo, los nodos no solo conservan su tipo estructural (élite o regular), sino que también incorporan información sobre su procedencia, lo que permite distinguir entre regiones del espacio de configuración exploradas de forma compartida o exclusiva. En consecuencia, se definen cinco nuevas categorías de nodos en la red fusionada:

- *shared-elite nodes*: Nodos compartidos por dos o más STN-i que contienen exclusivamente nodos clasificados como élite. Representan regiones de alta calidad comúnmente identificadas como relevantes por distintas ejecuciones del sintonizador.
- *shared-regular nodes*: Nodos presentes en múltiples STN-i que contienen únicamente nodos clasificados como regulares. Reflejan zonas del espacio exploradas repetidamente pero descartadas de forma consistente.
- *shared-mixed nodes*: Nodos compartidos por dos o más STN-i que contienen una mezcla de nodos clasificados como élite y regulares, sin cumplir el criterio para ser considerados como *shared-elite*. Capturan regiones del espacio donde existe desacuerdo entre ejecuciones respecto a su calidad.
- *network-elite nodes*: Nodos élite exclusivos de una única STN-i, etiquetados según la

configuración de *irace* que los generó. Representan regiones de calidad reconocidas solo por una ejecución particular.

- *network-regular nodes*: Nodos regulares exclusivos de una STN-i específica, también identificados por su procedencia. Corresponden a locaciones descartadas que solo fueron exploradas por una configuración de *irace*.

Adicionalmente, se conserva la clasificación topológica de los nodos (START, STANDARD, END), adaptada al contexto del modelo merged STN-i y aplicada exclusivamente a los nodos compartidos. En estos casos, la etiqueta topológica se asigna siguiendo una prioridad jerárquica: si al menos una de las STN-i originales clasifica la locación como END, el nodo fusionado adopta esta etiqueta; si no ocurre, pero alguna ejecución lo clasifica como START, se le asigna esta; y solo si ninguna ejecución lo marca como END ni START, se etiqueta como STANDARD. Esta convención garantiza la preservación de los puntos críticos del proceso de búsqueda y facilita la interpretación estructural de las trayectorias fusionadas.

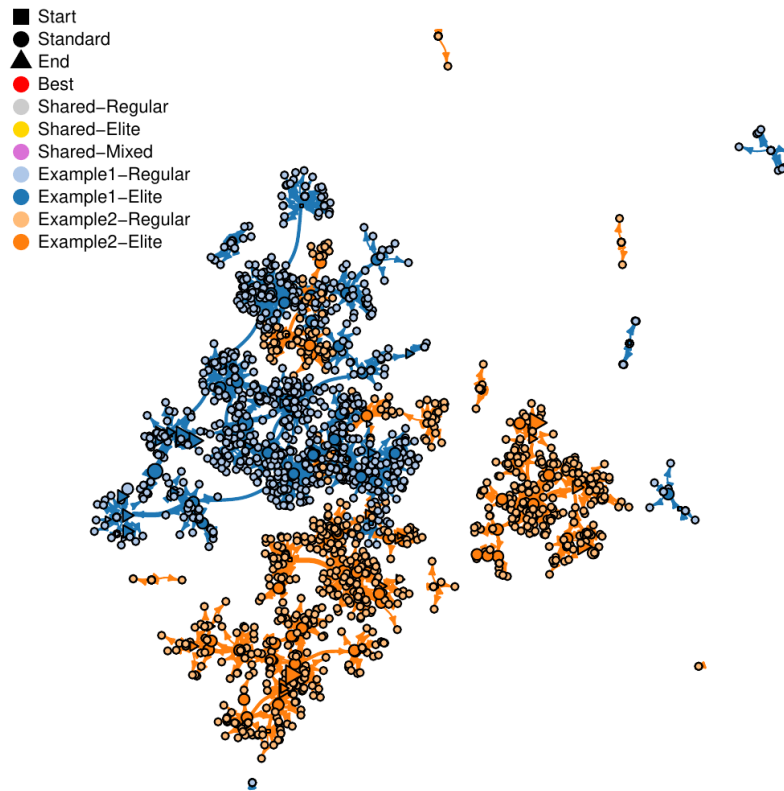


Figura 11: Ejemplo del modelo merged STN-i construido a partir de dos redes STN-i generadas de forma independiente sobre un mismo algoritmo, variando únicamente las condiciones de los hiperparámetros de *irace*.

Fuente: Elaboración propia.

La Figura 11 ilustra una instancia del modelo merged STN-i generada a partir de STN-i construidas con variaciones en los hiperparámetros de *irace*. Los nodos se distinguen por su tipo (élite o regular), procedencia (compartida o exclusiva) y rol topológico (START, STANDARD, END) mediante combinaciones de formas y colores indicadas en la parte superior izquierda.

Las regiones coloreadas en tonos anaranjado y azul agrupan nodos exclusivos de cada ejecución, mientras que los nodos compartidos —aquellos presentes en ambas redes STN-i— deberían destacarse en colores diferenciados según su tipo. Sin embargo, en este ejemplo particular, no se observa la presencia directa de nodos compartidos, ya sean élite, regulares o mixtos (*shared-elite*, *shared-regular*, *shared-mixed*), lo que indica una mínima superposición entre las trayectorias de búsqueda de ambas ejecuciones.

También se observan nodos START aislados, al igual que en el ejemplo de la Figura 10, correspondientes a configuraciones descartadas tempranamente, lo que refleja una amplia exploración inicial sobre el espacio de configuración. La estructura modular resultante sugiere una baja conectividad entre las dos redes, lo cual es coherente con la ausencia de nodos compartidos y refuerza la hipótesis de que las ejecuciones exploraron regiones diferenciadas del espacio, probablemente inducido por las diferencias en sus configuraciones experimentales y la baja cantidad de ejecuciones experimentales utilizadas.

3.1.2. Métricas de los modelos STN-i y merged STN-i

Con el objetivo de caracterizar las propiedades topológicas y funcionales de las redes construidas, se definen un conjunto de métricas que permiten analizar el comportamiento del proceso de búsqueda modelado por STN-i y su extensión merged STN-i. Estas métricas abarcan aspectos estructurales —como la conectividad, densidad y fragmentación de la red—, así como dinámicas propias del algoritmo *irace*, tales como la persistencia de configuraciones, la eficiencia de exploración y la convergencia hacia regiones de alta calidad. En conjunto, entregan una base cuantitativa para interpretar, comparar y diagnosticar el desempeño del sintonizador en distintos contextos experimentales.

- n : Número total de nodos presentes en la red. Representa la cantidad de locaciones que contienen configuraciones visitadas por *irace* durante el proceso de sintonización.
- $p_{n_{re}}$: Proporción de nodos clasificados como regulares. Corresponden a locaciones que no consideran ninguna configuración élite, por lo tanto, fueron descartadas en su iteración.
- $p_{n_{el}}$: Proporción de nodos clasificados como élite, calculada como la cantidad de nodos élite dividida por la cantidad total de nodos en la red. Indican las locaciones que contienen al menos una configuración seleccionada como élite, es decir, que sobrevivió a la etapa de descarte y tuvo impacto en iteraciones futuras.

- p_{nst} : Proporción de nodos etiquetados como START. Corresponden a las locaciones que contienen al menos una configuración generada en la primera iteración de irace, sin ninguna configuración clasificada como END. Pueden incluir además configuraciones STANDARD.
- p_{nsd} : Proporción de nodos etiquetados como STANDARD. Incluye locaciones que contienen únicamente configuraciones intermedias, es decir, ni de la primera ni de la última iteración, y sin ninguna marcada como END.
- p_{nen} : Proporción de nodos etiquetados como END. Son locaciones que contienen al menos una configuración generada en la última iteración del proceso; en tal caso, el nodo se clasifica como END independientemente de las demás configuraciones presentes.
- ed : Número total de arcos. Indica cuántas transiciones estructurales se produjeron entre nodos consecutivos debido a relaciones de descendencia entre configuraciones.
- p_{edw} : Proporción de arcos que conectan un nodo con otro de menor calidad. Representan transiciones con empeoramiento, asociadas a decisiones divergentes en la búsqueda.
- p_{edeq} : Proporción de arcos que conectan nodos de igual calidad. Corresponden a transiciones neutras sin impacto en la calidad de la solución.
- p_{edimp} : Proporción de arcos que conectan un nodo con otro de mejor calidad. Representan mejoras en la trayectoria de búsqueda.
- $tend_{ed}$: Tendencia normalizada de arcos con mejora respecto a las que empeoran. Se calcula como la cantidad de arcos que conectan nodos de mejor calidad dividida por la suma de los arcos que representan mejoras y empeoramientos. Valores altos indican una búsqueda orientada hacia configuraciones de mayor calidad, mientras que valores bajos reflejan una tendencia a transiciones hacia soluciones de menor calidad. Un valor cercano a 0.5 sugiere un comportamiento equilibrado.
- p_{c-nre} : Proporción de configuraciones etiquetadas como regulares, considerando todas las configuraciones agrupadas en locaciones de la red. Permite evaluar la presión global de descarte impuesta por irace.
- p_{c-nel} : Proporción de configuraciones etiquetadas como élite, considerando todas las configuraciones agrupadas en locaciones de la red. Mide el peso estructural del conjunto élite durante el proceso de sintonización.
- p_{c-nre}^{st} : Proporción de configuraciones regulares agrupadas en nodos START, respecto al total de configuraciones contenidas en dichos nodos. Indica qué fracción de configuraciones generadas inicialmente fue descartada.
- p_{c-nel}^{st} : Proporción de configuraciones élite agrupadas en nodos START, respecto al total de configuraciones en dichos nodos. Refleja la calidad inicial del proceso de muestreo.

- $tend_c$: Tendencia de configuraciones que sobreviven entre iteraciones. Se calcula como el promedio, por ejecución independiente de *irace*, de la proporción de configuraciones que pasan de una iteración a la siguiente, y luego promediando este valor sobre todas las ejecuciones. Refleja la estabilidad del proceso de descarte.
- $p_{n_{bt}}$: Proporción de nodos asociados a las mejores soluciones. Se calcula como la cantidad de nodos que contienen configuraciones con mejor calidad (según el tipo de problema, minimización o maximización) sobre el total de nodos en la red. Evalúa la cobertura del proceso sobre regiones de alta calidad.
- av_n^{in}, av_n^{out} : Grado promedio de arcos de entrada y salida considerando todos los nodos. Miden la conectividad global y la direccionalidad media del flujo de búsqueda.
- $av_{n_{re}}^{in}, av_{n_{re}}^{out}$: Grado promedio de arcos de entrada y salida para los nodos regulares. Indican el nivel de conexión de las regiones descartadas y su integración estructural.
- $av_{n_{el}}^{in}, av_{n_{el}}^{out}$: Grado promedio de arcos de entrada y salida para los nodos élite. Reflejan la centralidad estructural de las regiones influyentes.
- $av_{n_{bt}}^{in}, av_{n_{bt}}^{out}$: Grado promedio de arcos de entrada y salida para los nodos asociados a las mejores configuraciones encontradas. Reflejan el nivel de atracción estructural hacia regiones óptimas y su papel como destinos o puntos de referencia en las trayectorias exploradas.
- $st_{n_{bt}}^{in}$: Fuerza promedio de entrada para los nodos asociados a las mejores configuraciones encontradas, normalizada por el número de ejecuciones independientes. Esta métrica refleja la intensidad acumulada de las trayectorias que convergen hacia estas soluciones óptimas, considerando la frecuencia con la que son alcanzadas desde otras configuraciones en las ejecuciones.
- len_{pt} : Longitud promedio de los caminos más cortos desde nodos START hasta los mejores nodos. Indica la eficiencia estructural del recorrido exploratorio.
- pt : Número total de trayectorias óptimas hacia nodos de alta calidad. Estima la accesibilidad de regiones prometedoras desde los puntos de inicio.
- co : Número de componentes conexas (excluye nodos regulares de inicio no conectados). Permite analizar la fragmentación de la red y la existencia de trayectorias aisladas, que en el contexto de *irace* reflejan estructuras inactivas o no influyentes, como la generación inicial de nodos regulares.
- $p_{n_{sh}}$: Para merged STN-i, proporción general de nodos compartidos entre dos o más redes respecto al total de nodos. Esta métrica agrupa todos los tipos de nodos compartidos (*shared-regular*, *shared-elite* y *shared-mixed*), y mide el grado de convergencia estructural entre las trayectorias exploradas por distintas ejecuciones de *irace*.
- $p_{n_{shre}}$: Para merged STN-i, proporción de *shared-regular nodes*. Refleja regiones descartadas que fueron comúnmente visitadas por distintas redes.

- $p_{n_{shel}}$: Para merged STN-i, proporción de *shared-elite nodes*. Indica locaciones que contienen configuraciones élite en todas las STN-i en las que aparecen, representando regiones de consenso de alta calidad.
- $p_{n_{shmex}}$: Para merged STN-i, proporción de *shared-mixed nodes*. Representa regiones con discrepancia en la evaluación estructural entre diferentes configuraciones del sintonizador.
- $av_{n_{sh}}^{in}, av_{n_{sh}}^{out}$: Para merged STN-i, grado promedio de arcos de entrada y salida de todos los nodos compartidos. Mide su participación estructural dentro de la red fusionada, indicando qué tan frecuentemente estas locaciones comunes son utilizadas como origen o destino de transiciones.
- $av_{n_{shre}}^{in}, av_{n_{shre}}^{out}$: Para merged STN-i, grado promedio de arcos de entrada y salida de los *shared-regular nodes*. Evalúa su función estructural como posibles puntos de paso, retorno o terminación en trayectorias comunes descartadas.
- $av_{n_{shel}}^{in}, av_{n_{shel}}^{out}$: Para merged STN-i, grado promedio de arcos de entrada y salida de los *shared-elite nodes*. Permite estimar la centralidad e influencia global de los nodos élite como puntos clave dentro del proceso de generación de nuevas configuraciones.
- $av_{n_{shmex}}^{in}, av_{n_{shmex}}^{out}$: Para merged STN-i, grado promedio de arcos de entrada y salida de los *shared-mixed nodes*. Estas métricas reflejan la centralidad de regiones compartidas que presentan ambigüedad en su rol estructural, revelando posibles puntos de transición o evaluación variable entre ejecuciones.

3.2. Algoritmo *irace-STN-i*: Extracción e interpretación de trayectorias de *irace*

Una vez definido el modelo STN-i, se introduce el algoritmo *irace-STN-i*, encargado de transformar los resultados de múltiples ejecuciones independientes de *irace* en una estructura compatible con dicho modelo. Su objetivo es representar de forma explícita las trayectorias exploradas durante el proceso de sintonización automática en el espacio discretizado de parámetros, capturando tanto la dinámica iterativa como las relaciones de descendencia entre configuraciones.

El proceso se divide en cuatro etapas principales. En la primera, se definen los parámetros de entrada que controlan la interpretación del espacio de búsqueda y la generación del archivo de salida. Luego, se interpreta el archivo de parámetros y se extraen las configuraciones evaluadas desde los archivos `.Rdata`, construyendo estructuras intermedias por iteración y por locación. En la tercera etapa, se generan las trayectorias conectando configuraciones entre iteraciones, distinguiendo entre creaciones, descendencias válidas y persistencias. Finalmente, se construye un archivo con formato tabular que registra, para cada arco de la red,

los atributos de ambas locaciones involucradas, permitiendo su posterior análisis mediante el modelo STN-i.

3.2.1. Etapa 1: Datos de entrada

Esta etapa define los insumos necesarios para ejecutar el algoritmo *irace-STN-i*, especificados mediante argumentos por línea de comandos. Los parámetros principales son:

- `--input`: Directorio que contiene los archivos `.Rdata` generados por *irace*. Cada archivo representa una ejecución independiente bajo el mismo escenario.
- `--parameters`: Ruta al archivo `.csv` que define los parámetros del espacio de búsqueda, incluyendo tipo, dominio y datos de codificación de locación cada uno.
- `--output`: Carpeta donde se almacenará el archivo de salida generado.
- `--criteria`: Operador utilizado para seleccionar el valor de calidad representativo de cada locación. Las opciones válidas son `min`, `max`, `mean`, `median` o `mode` (por defecto: `min`).
- `--significance`: Número de cifras decimales usadas para redondear los valores de calidad (por defecto: 2).
- `--output_file` (opcional): Nombre del archivo de salida a generar (por defecto: `stn_file.stn`).

Formato del archivo de parámetros El archivo debe estar separado por punto y coma (;) y contener las siguientes columnas:

- **NAME**: Nombre del parámetro.
- **CONDITIONAL**: Booleano que indica si el parámetro depende de otro.
- **TYPE**: Tipo del parámetro: `c` (categórico), `o` (ordinal), `i` (entero), `r` (real).
- **VALUES_ARRAY**: Para `c` y `o`, es la lista de todos valores posibles. Para `i` y `r`, es el rango de valores permitidos (por ejemplo, `(0,5)`).
- **LOCATIONS_ARRAY**: Para parámetros `c` y `o`, define un mapeo explícito entre valores y códigos de locación. Para `i` y `r`, contiene un par (`step`, `significance`) que indica el nivel de discretización y precisión.

Este archivo es esencial para codificar de forma única las configuraciones evaluadas por *irace*, permitiendo construir una representación estructurada del proceso de búsqueda en el modelo STN-i.

3.2.2. Etapa 2: Lectura de datos

Esta etapa se encarga de interpretar el archivo de parámetros y los resultados de ejecuciones de `irace` para organizar la información requerida por el modelo STN-i.

Primero, se procesa cada parámetro del archivo `.csv`, construyendo una estructura de dominios que permite codificar configuraciones en locaciones discretas. Según el tipo de parámetro:

- `c` y `o`: Se mapean mediante diccionarios explícitos de categorías a valores numéricos.
- `i` y `r`: Se discretizan usando el `step` y la `significancy` especificada.
- Parámetros condicionales no activos: Se codifican como subcadenas de 'X', manteniendo la longitud total del código de locación.

Luego, se leen los archivos `.Rdata` desde la carpeta indicada, donde cada archivo representa una ejecución con distinta semilla. Por cada configuración evaluada, se extraen el código de locación generado a partir de sus valores discretizados, el tipo de nodo (START, STANDARD o END) según su posición en la trayectoria, y su pertenencia al conjunto ELITE.

Con esta información se construyen dos estructuras principales: un diccionario por iteración y ejecución que contiene los datos de cada configuración evaluada, y un diccionario global de locaciones que acumula, para cada código de locación, las calidades observadas, el tipo de nodo predominante y el estado final ELITE.

Dado que una misma locación puede ser compartida por múltiples configuraciones, se resuelve su tipo predominante aplicando una prioridad fija: STANDARD < START < END. Esto asegura una asignación coherente en función del rol estructural observado.

Luego, se calcula el valor de calidad representativo por locación según el criterio definido en `criteria`, redondeado con `significancy`. Estos atributos consolidan la información necesaria para construir las trayectorias.

3.2.3. Etapa 3: Construcción de trayectorias

Con la información organizada, el algoritmo construye las trayectorias de búsqueda realizadas por `irace`, modeladas como arcos entre locaciones consecutivas dentro de cada ejecución. Cada configuración evaluada es codificada como una locación única, etiquetada con su tipo (START, STANDARD, END) y su estatus (ELITE o REGULAR).

La conexión entre configuraciones se define según reglas específicas relacionadas con su iteración y estado. En la primera iteración, tanto las configuraciones élite como las no élite

se conectan consigo mismas mediante una auto-referencia. Esta conexión representa que no tienen un padre previo, es decir, que son configuraciones iniciales. Para reflejar que estas auto-referencias no forman parte activa de las trayectorias de búsqueda, se etiquetan con `PATH = FALSE`, evitando considerarlas como parte del seguimiento iterativo.

En iteraciones posteriores a la primera, si una configuración posee un `PARENT` válido correspondiente a una configuración de la iteración inmediatamente anterior, se crea un arco dirigido desde dicho padre hacia la configuración actual, estableciendo así la continuidad en el proceso de generación y evolución de configuraciones.

Sin embargo, si en estas iteraciones superiores a la primera una configuración élite no tiene un padre válido en la iteración previa, esto indica que dicha configuración ha sobrevivido el proceso de descarte. Para reflejar esta supervivencia, se crea una auto-referencia desde la configuración hacia sí misma, que en este caso se etiqueta con `PATH = TRUE`, señalando que forma parte activa de la trayectoria, ya que representa la persistencia estructural del conjunto élite a lo largo de las iteraciones.

Cada arco generado almacena información completa sobre las locaciones involucradas, lo que permite consolidar la red de trayectorias en la siguiente etapa del proceso.

3.2.4. Etapa 4: Generación de archivo de salida

Finalizada la construcción de trayectorias, se genera un archivo `stn_file.txt` en formato de texto plano con campos separados por tabulaciones. Este archivo consolida todas las conexiones entre locaciones observadas en las ejecuciones de `irace`. Cada fila representa un arco (o auto-referencia), registrando los atributos de ambos nodos involucrados.

El archivo se guarda automáticamente en el directorio especificado por el parámetro `output_folder` y puede ser utilizado directamente para análisis o visualización mediante herramientas del modelo STN-i. Cada fila del archivo incluye las siguientes columnas:

- **Run:** Identificador de la ejecución numérico asignado a la ejecución independiente.
- **Path:** Identificador de si una conexión entre 2 locaciones debe ser considerado o no como parte de la trayectoria.
- **Solution1, Solution2:** Códigos de locación de los nodos origen y destino.
- **Fitness1, Fitness2:** Valor de calidad representativo de cada locación, calculado según criterios y redondeado con `significancy`.
- **Elite1, Elite2:** Estado de elite consolidado por locación.
- **Original_Elite1, Original_Elite2:** Estado de elite de la configuración puntual evaluada.

- Type1, Type2: Tipo de nodo final por locación.
- Original_Type1, Original_Type2: Tipo observado de la configuración puntual (START, STANDARD, END).
- Iteration1, Iteration2: Iteración donde aparece cada configuración en la trayectoria.

A continuación, se muestra un ejemplo ilustrativo con las primeras filas del archivo generado, reflejando distintas transiciones entre configuraciones evaluadas durante ejecuciones independientes:

Run	Path	Fitness1	Solution1	Elite1	Original_elite1	Type1	Original_type1	Iteration1	Fitness2	Solution2	Elite2	Original_elite2	Type2	Original_type2	Iteration2
1	FALSE	10.50	0125Y	REGULAR	REGULAR	STANDARD	STANDARD	1	10.50	0125Y	REGULAR	REGULAR	STANDARD	STANDARD	1
1	TRUE	12.34	0123X	ELITE	ELITE	START	START	1	15.67	0124X	REGULAR	REGULAR	END	STANDARD	2
2	TRUE	11.00	0126Z	ELITE	ELITE	STANDARD	STANDARD	2	13.20	0127Z	ELITE	ELITE	END	END	3
2	TRUE	14.00	0128X	ELITE	ELITE	END	END	2	14.50	0129X	ELITE	ELITE	END	END	3

Este ejemplo permite observar distintos casos frecuentes en la estructura de las STN-i. La primera fila muestra una auto-referencia (Path = FALSE) correspondiente a una configuración regular descartada en la misma iteración donde fue generada. La segunda fila ilustra una transición activa (Path = TRUE) entre una configuración élite inicial y una configuración regular posterior. Las filas siguientes muestran trayectorias activas mantenidas por configuraciones élite, lo que refleja su persistencia e influencia en iteraciones sucesivas del proceso de sintonización.

3.3. Conclusiones

La propuesta desarrollada en este capítulo establece un marco formal y operativo para modelar el comportamiento exploratorio de *irace* mediante el enfoque de *Search Trajectory Networks*.

El modelo STN-i permite representar explícitamente la dinámica iterativa del proceso de sintonización, distinguiendo entre configuraciones descartadas y seleccionadas, y resaltando cómo estas últimas dan forma a las trayectorias de búsqueda. La tipificación estructural de los nodos (por rol topológico y estado élite) facilita el análisis de patrones como bifurcaciones, convergencias o persistencias en el espacio de configuración.

El algoritmo *irace-STN-i*, por su parte, automatiza la extracción y estructuración de trayectorias a partir de los resultados generados por *irace*. Su diseño modular garantiza la trazabilidad de la información, desde la codificación de parámetros hasta la generación de los arcos que conforman la red.

Asimismo, la extensión merged STN-i habilita el análisis comparativo entre múltiples ejecuciones del sintonizador, identificando regiones exploradas en común, zonas exclusivas y puntos de convergencia estructural. Las métricas definidas para ambos modelos permiten

evaluar propiedades clave de la búsqueda, como su cobertura, eficiencia, calidad alcanzada y diversidad.

En conjunto, esta propuesta entrega una base estructural sólida para interpretar el comportamiento interno de *irace*, complementando los análisis clásicos centrados en el rendimiento y abriendo nuevas posibilidades para su diagnóstico, ajuste y mejora fundamentada.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Este capítulo presenta el proceso de validación de la solución propuesta, centrada en la aplicación del modelo STN-i —y su variante fusionada, merged STN-i— sobre las trayectorias de búsqueda generadas por el sintonizador automático *irace*. El objetivo es analizar estructuralmente el comportamiento exploratorio de *irace* durante el proceso de configuración de parámetros en algoritmos metaheurísticos, considerando distintos dominios de aplicación, niveles de presupuesto y mecanismos de diversificación.

La validación se organiza en cuatro secciones principales. En primer lugar, se define el escenario experimental, donde se especifican los algoritmos objetivo, los conjuntos de instancias utilizados, las locaciones construidas para representar el espacio de configuraciones y las combinaciones de hiperparámetros empleadas en *irace*. Luego, se presentan los resultados obtenidos a partir de las ejecuciones realizadas, incluyendo visualizaciones y métricas estructurales de las redes generadas por STN-i para cada caso de estudio. A continuación, se realiza un análisis detallado de los resultados, comparando configuraciones y evaluando el impacto de distintas condiciones sobre la dinámica de exploración de *irace*. Finalmente, se exponen breves conclusiones que sintetizan los hallazgos más relevantes de esta sección, destacando el valor del enfoque STN-i para la interpretación estructural de procesos de sintonización automática.

Este enfoque metodológico permite evaluar no solo la calidad de las configuraciones obtenidas, sino también las trayectorias recorridas por el algoritmo durante la búsqueda, aportando una perspectiva complementaria a los enfoques tradicionales basados exclusivamente en métricas de desempeño.

4.1. Escenario experimental

Esta sección presenta el entorno experimental diseñado para evaluar el proceso de sintonización automática de parámetros de *irace* mediante el modelo STN-i. Primero, se describen los algoritmos objetivo, es decir, los algoritmos metaheurísticos cuyos parámetros serán optimizados. Luego, se detallan los conjuntos de instancias utilizados en la evaluación, así como la definición de las locaciones necesarias para el análisis estructural a través de STNs-i. Finalmente, se especifican los hiperparámetros de *irace* empleados durante el proceso de sintonización, junto con la metodología utilizada para validar la calidad de las configuraciones resultantes.

4.1.1. Algoritmos objetivo

En este estudio se consideran tres algoritmos metaheurísticos aplicados a problemas combinatorios clásicos. La calidad de sus soluciones depende críticamente de la configuración de sus parámetros, por lo que se emplea *irace* para su sintonización. A continuación, se detallan tanto los parámetros a optimizar como los componentes internos fijos que definen su dinámica de búsqueda.

MMASQAP [Stützle y Hoos, 1998] Es una adaptación del algoritmo *MAX-MIN Ant System* (MMAS) al problema de asignación cuadrática (QAP). Como variante avanzada de ACO, MMAS introduce límites explícitos sobre los valores de feromonas para evitar la convergencia prematura, reforzando únicamente las mejores soluciones. Esta estrategia fomenta un balance entre exploración y explotación, crucial para abordar eficientemente espacios de búsqueda combinatorios.

Parámetro	Tipo	Dominio
<code>localsearch</code>	c	$\{0, 1, 2, 3\}$
ρ	r	$[0.01, 1.00]$
<code>ants</code>	i	$[1, 100]$

Tabla 1: Parámetros que serán optimizados para MMASQAP.

En la Tabla 1 se presentan los parámetros considerados para la sintonización automática del algoritmo. El parámetro `localsearch` define el procedimiento de búsqueda local aplicado, donde los valores posibles corresponden a: 0 para primera mejora con *don't look bits*¹⁷, 1 para primera mejora sin dicha técnica, 2 para mejor mejora, y 3 para Tabu Search con longitud $4n$.

El coeficiente ρ regula la evaporación de feromonas, es decir, el grado en que las decisiones pasadas pierden influencia. Su ajuste afecta directamente el balance exploración-explotación. Por otro lado, `ants` define cuántas soluciones se generan por iteración; un mayor valor implica una exploración más amplia, pero a mayor costo computacional.

Además de los parámetros ajustados, MMASQAP se ejecuta con hiperparámetros fijos que definen su configuración base. Se establece `trials` = 1 para asegurar una única ejecución por configuración, mientras que `iterations` = 1 limita el proceso a una sola iteración del algoritmo. Adicionalmente, se fija `evaluations` = 5000 como presupuesto computacional máximo, expresado en número de evaluaciones de soluciones individuales. Este último parámetro representa una extensión respecto a la versión original del algoritmo [Stützle y Hoos, 1998], incorporada para controlar el esfuerzo computacional de manera

¹⁷Técnica que evita revisar ciertos movimientos durante la búsqueda local si se considera que es poco probable que mejoren la solución actual, con el objetivo de reducir el tiempo de cómputo sin afectar significativamente la calidad.

explícita. Todos estos valores se mantienen constantes durante los experimentos, permitiendo que la comparación entre configuraciones se base exclusivamente en los parámetros sujetos a sintonización.

ACOTSP [Stützle, 2002] Es un paquete de software que implementa múltiples variantes de algoritmos ACO aplicados al TSP. ACOTSP permite experimentar con estrategias como AS, MMAS, EAS, RAS y ACS, y contempla tanto parámetros comunes como condicionales, en función del algoritmo base y la configuración de búsqueda local seleccionada.

Parámetro	Tipo	Dominio	Condición
algorithm	<i>c</i>	$\{as, mmass, eas, ras, acs\}$	-
localsearch	<i>c</i>	$\{0, 1, 2, 3\}$	-
α	<i>r</i>	[0.00, 5.00]	-
β	<i>r</i>	[0.00, 10.00]	-
ρ	<i>r</i>	[0.01, 1.00]	-
ants	<i>i</i>	[5, 100]	-
q_0	<i>r</i>	[0.00, 1.00]	algorithm = <i>acs</i>
rasrank	<i>i</i>	[1, 100]	algorithm = <i>ras</i>
elitistants	<i>i</i>	[1, 750]	algorithm = <i>eas</i>
nnls	<i>i</i>	[5, 50]	localsearch $\in \{1, 2, 3\}$
dlb	<i>c</i>	$\{0, 1\}$	localsearch $\in \{1, 2, 3\}$

Tabla 2: Parámetros a sintonizar para ACOTSP.

En la Tabla 2 se presentan los parámetros considerados para la sintonización automática del algoritmo. Los parámetros α y β controlan el peso relativo entre la información de feromona y la heurística durante la construcción de soluciones. El parámetro ρ representa la tasa de evaporación de feromonas. *ants* define la cantidad de hormigas por iteración.

Los parámetros q_0 , *rasrank* y *elitistants* son específicos de ciertos algoritmos: q_0 aplica solo a ACS y determina la probabilidad de aplicar una política de explotación; *rasrank* es exclusivo de RAS y controla la cantidad de hormigas cuyos *tours* influyen en la actualización de feromonas; *elitistants* está definido solo para EAS y regula cuántas veces el mejor *tour* refuerza las feromonas.

Respecto a la búsqueda local, el parámetro *localsearch* define el método empleado (0: sin búsqueda local, 1: 2-opt, 2: 2.5-opt, 3: 3-opt). Si se selecciona una opción distinta de 0, se habilitan también *nnls*, que determina el tamaño de la vecindad restringida, y *dlb*, un parámetro binario que activa la optimización mediante la técnica *don't look bits*.

Además de los parámetros sintonizados, ACOTSP se ejecuta con hiperparámetros fijos que definen su comportamiento base: se establece *tries* = 1 para limitar la evaluación a una única ejecución por configuración, y *tours* = 3000 como presupuesto computacional total, expresado en el número máximo de *tours* globales generados por las hormigas durante

la ejecución. Estos valores se mantienen constantes a lo largo de todos los experimentos, asegurando que las diferencias observadas provengan únicamente de los parámetros optimizados.

PSO-X [Camacho-Villalón et al., 2022] Es un *framework* orientado a la construcción automática de variantes del algoritmo PSO. Este *framework* propone una arquitectura modular que permite definir explícitamente los componentes de un algoritmo PSO, facilitando su experimentación, extensión y adaptación a distintas clases de problemas. En este trabajo se emplea una configuración específica del framework que reproduce el comportamiento de la variante estándar SPSO 2011 [Zambrano-Bigiarini et al., 2013], denominada SPS011 dentro de PSO-X. Esta versión se caracteriza por una actualización de velocidades basada en una interpretación geométrica de los desplazamientos, eliminando dependencias cartesianas y promoviendo una mayor diversidad poblacional durante la búsqueda.

Parámetro	Tipo	Dominio	Condición
particles	i	[2, 200]	-
topology	c	{0, 1, 2, 3, 4, 5, 6}	-
modInfluence	c	{0, 1, 2}	-
branching	i	[4, 20]	topology = 6
tSchedule	i	[2, 10]	topology = 5
phi1	r	[0.00, 2.50]	-
phi2	r	[0.00, 2.50]	-
inertia	r	[0.00, 0.90]	-

Tabla 3: Parámetros a sintonizar para PSO-X.

En la Tabla 3 se detallan los parámetros sometidos a sintonización. El parámetro *particles* determina la cantidad de partículas en el enjambre, influyendo en la capacidad de exploración inicial y en el costo computacional por iteración. *topology* define la estructura de vecindad, con variantes como anillo, estrella, árbol jerárquico o enfriamiento progresivo. El parámetro *modInfluence* modula la intensidad de la influencia entre partículas según la topología seleccionada.

Los coeficientes *phi1* (cognitivo) y *phi2* (social) regulan la atracción de cada partícula hacia su mejor posición histórica y hacia la mejor posición de su vecindad. El parámetro *inertia* controla el peso de la velocidad previa en la nueva actualización, influyendo en la estabilidad y convergencia del algoritmo. Los parámetros *branching* y *tSchedule* son condicionales: el primero se aplica exclusivamente cuando *topology* = 6, estableciendo la cantidad de ramas jerárquicas; el segundo se activa cuando *topology* = 5, modulando la tasa de cambio en la estructura de conectividad.

Además de los parámetros optimizados, PSO-X se ejecuta con un conjunto fijo de hiperparámetros que definen el comportamiento base del algoritmo. En primer lugar, se desactiva el uso de poblaciones cooperativas secundarias mediante *populationCS* = 0, y se emplea

una inicialización aleatoria estándar de las partículas con `pInitType = 0`. Los coeficientes internos que regulan la actualización de velocidades se fijan como `accelCoeffCS = 0`, `omega1CS = 0`, `omega2CS = 2` y `omega3CS = 1`, valores que replican la dinámica geométrica característica de SPSO11. Asimismo, se activa la política de selección dinámica del mejor vecino (*Dynamic Nearest Personal Best*) mediante `DNPP = 1`, y se desactivan los operadores de perturbación adicionales con `perturbation1 = 0` y `perturbation2 = 0`. Para mantener la rotación estocástica en el espacio de búsqueda, se habilita el uso de matrices ortonormales aleatorias con `randomMatrix = 1`. Finalmente, el presupuesto computacional se define en función del número máximo de evaluaciones mediante `evaluations = -3`, desactivando cualquier límite explícito por iteraciones con `iterations = -1`.

Estos valores permanecen constantes en todos los experimentos, garantizando que las variaciones observadas en el rendimiento y la dinámica del algoritmo se deban exclusivamente a los parámetros sometidos a sintonización. De este modo, se asegura la estabilidad estructural del modelo base y se facilita el análisis comparativo entre configuraciones generadas automáticamente.

4.1.2. Instancias

Los conjuntos de instancias considerados en este estudio son utilizados exclusivamente para el proceso de entrenamiento del sintonizador `irace`. A continuación, se describen los diferentes conjuntos empleados según el tipo de problema y dominio:

- **QAP-60:** Conjunto compuesto por 60 instancias del QAP, cada una con 60 elementos y distintos niveles de dispersión. Se incluyen 30 instancias *Euclidean Structured* con un valor de dispersión de 0.90 y 30 instancias *Random Random* con dispersión 0.00. Estas instancias fueron introducidas por Hussin y Stützle [Saifullah Hussin y Stützle, 2014].
- **TSP-2000:** Conjunto compuesto por 400 instancias *Random Uniform Euclidean* del TSP, cada una con 2000 ciudades. Este conjunto se caracteriza por su alta homogeneidad estructural. Las instancias fueron empleadas originalmente en el artículo de López-Ibáñez et al. [López-Ibáñez et al., 2016].
- **PSO-Mix:** Conjunto compuesto por funciones *híbridas y de composición*, que combinan múltiples subfunciones con estructuras diversas. Incluye funciones de CEC'05 (F15–F25), CEC'14 (F14–F19) y SOCO'10 (F9–F18), todas con 30 dimensiones. Estas funciones presentan paisajes de optimización altamente complejos. Las definiciones se encuentran en [Suganthan et al., 2005, Liang et al., 2013, Herrera et al., 2010].
- **PSO-Mul:** Conjunto de funciones *multimodales*, que incluyen múltiples óptimos globales y locales. Las funciones provienen de las competiciones CEC'05 (F6–F14), CEC'14 (F4–F13) y SOCO'10 (F5–F8), todas con 30 dimensiones. Este conjunto está diseñado para evaluar la capacidad de exploración global de los algoritmos. Las definiciones corresponden a [Suganthan et al., 2005, Liang et al., 2013, Herrera et al., 2010].

- **PSO-Uni:** Conjunto compuesto por funciones *unimodales* tomadas desde competencias estándar de optimización continua. Incluye funciones de CEC'05 (F1–F5), CEC'14 (F1–F3) y SOCO'10 (F1–F4), abarcando problemas con un único óptimo global y distintas transformaciones (rotaciones, desplazamientos). Todas las funciones tienen 30 dimensiones y provienen de las definiciones oficiales de [Suganthan *et al.*, 2005, Liang *et al.*, 2013, Herrera *et al.*, 2010].

4.1.3. Locaciones

Para cada algoritmo objetivo se definen tres tamaños de locaciones L_1 , L_2 y L_3 , orientados a establecer distintos niveles de granularidad en el espacio de parámetros numéricos. Estas locaciones se construyen exclusivamente sobre parámetros numéricos, es decir, de tipo i (entero) y r (real), excluyendo parámetros categóricos (c) y ordinales (o), los cuales conservan sus dominios originales debido a su carácter no cuantitativo o no ordenado.

La dimensión base discretizada $|D|$ de cada parámetro se calcula a partir del rango definido por los valores mínimo y máximo, utilizando una resolución mínima s que corresponde a 1 para parámetros enteros y a 0.01 para parámetros reales, considerando una precisión de dos decimales:

$$|D| = \left\lceil \frac{\max - \min}{s} \right\rceil + 1 \quad (12)$$

Esta fórmula entrega el número total de valores distintos que puede tomar un parámetro bajo dicha resolución mínima, incluyendo el valor mínimo. A partir de este dominio base, se define el tamaño de cada locación L_i ajustando el paso s_i para lograr una granularidad específica. El tamaño resultante de una locación se estima como:

$$|D_{L_i}| \approx \left\lceil \frac{\max - \min}{s_i} \right\rceil \quad (13)$$

donde s_i representa el paso efectivo aplicado, distinto de s en función del nivel de granularidad buscado. En este trabajo, se establece como criterio que las locaciones L_1 , L_2 y L_3 correspondan aproximadamente a divisiones del dominio base $|D|$ en 5, 15 y 25 partes respectivamente, es decir:

$$|D_{L_i}| \approx \left\lceil \frac{|D|}{k_i} \right\rceil \quad \text{con } k_i \in \{5, 15, 25\} \quad (14)$$

Este enfoque permite representar de forma escalonada y controlada la granularidad del espacio de búsqueda, facilitando un análisis multiescala de las trayectorias. Locaciones más

finas como L_1 mantienen una mayor resolución que permite identificar patrones locales y transiciones sutiles, mientras que locaciones más gruesas como L_3 simplifican el espacio para destacar comportamientos globales o tendencias generales en el proceso de configuración.

En MMASQAP, todos los parámetros numéricos se discretizan siguiendo esta estrategia. La Tabla 4 presenta el paso mínimo s , el tamaño original del dominio, los valores de s_i para cada locación y sus tamaños discretizados. Los parámetros categóricos, como `localsearch`, conservan sus dominios sin alteración.

Parámetro	s	$ D $	s_1	$ D_{L_1} $	s_2	$ D_{L_2} $	s_3	$ D_{L_3} $
<code>localsearch</code>	-	4	-	4	-	4	-	4
ρ	0.01	100	0.05	20	0.15	7	0.25	4
<code>ants</code>	1	100	5	20	15	7	25	4

Tabla 4: Discretización de parámetros por locación en MMASQAP.

En ACOTSP se aplica el mismo enfoque sobre todos los parámetros numéricos, sin distinción entre parámetros globales o condicionales. Como se muestra en la Tabla 5, los parámetros categóricos conservan sus dominios completos, mientras que el resto se discretiza mediante los pasos adaptados a cada locación.

Parámetro	s	$ D $	s_1	$ D_{L_1} $	s_2	$ D_{L_2} $	s_3	$ D_{L_3} $
<code>algorithm</code>	-	5	-	5	-	5	-	5
<code>localsearch</code>	-	4	-	4	-	4	-	4
α	0.01	501	0.05	101	0.15	34	0.25	21
β	0.01	1001	0.05	201	0.15	67	0.25	41
ρ	0.01	100	0.05	20	0.15	7	0.25	4
<code>ants</code>	1	96	5	20	15	7	25	4
$q0$	0.01	100	0.05	20	0.15	7	0.25	4
<code>rasrank</code>	1	100	5	20	15	7	25	4
<code>elitistants</code>	1	750	5	150	15	50	25	30
<code>nnls</code>	1	46	5	10	15	4	25	2
<code>dlb</code>	-	2	-	2	-	2	-	2

Tabla 5: Discretización de parámetros por locación en ACOTSP.

En PSO-X se utiliza la misma estrategia, aplicando la discretización sobre todos los parámetros numéricos definidos. La Tabla 6 presenta los tamaños de dominio resultantes para cada locación, destacando el ajuste progresivo de la granularidad para permitir exploraciones jerárquicas de diferente resolución.

Parámetro	s	$ D $	s_1	$ D_{L_1} $	s_2	$ D_{L_2} $	s_3	$ D_{L_3} $
particles	1	199	5	40	15	14	25	8
topology	-	7	-	7	-	7	-	7
modInfluence	-	3	-	3	-	3	-	3
branching	1	17	5	4	15	2	17	1
tSchedule	1	8	5	2	8	1	8	1
phi1	0.01	251	0.05	51	0.15	17	0.25	11
phi2	0.01	251	0.05	51	0.15	17	0.25	11
inertia	0.01	91	0.05	19	0.15	7	0.25	4

Tabla 6: Discretización de parámetros por locación en PSO-X.

4.1.4. Hiperparámetros de irace

Cada ejecución de *irace* se configura a partir de combinaciones específicas de hiperparámetros, definidas en función de dos dimensiones clave: el presupuesto computacional disponible y el uso del mecanismo de *soft-restart*. Estas combinaciones permiten analizar cómo varía el comportamiento del proceso de sintonización bajo distintos niveles de presión exploratoria y estrategias de diversificación.

El presupuesto se asigna proporcionalmente a la cantidad de parámetros sintonizables de cada algoritmo objetivo, considerando dos niveles: 250 y 500. Por ejemplo, MMASQAP, que contempla tres parámetros, recibe presupuestos de 750 y 1500 evaluaciones, respectivamente. Esta asignación proporcional busca garantizar un uso equilibrado de recursos computacionales, manteniendo condiciones comparables entre algoritmos de distinta complejidad.

Además, se contempla la activación opcional del mecanismo de *soft-restart*, el cual reinicia parcialmente la población de configuraciones cuando no se observan mejoras significativas durante varias iteraciones. Esta estrategia busca mantener la diversidad exploratoria y prevenir la convergencia prematura.

Combinando los dos niveles de presupuesto con las dos condiciones posibles respecto al uso de *soft-restart* (activado o desactivado), se definen cuatro configuraciones experimentales por algoritmo objetivo:

- BL-SR: presupuesto bajo (250) con *soft-restart* activado,
- BL-WSR: presupuesto bajo sin *soft-restart*,
- BH-SR: presupuesto alto (500) con *soft-restart* activado,
- BH-WSR: presupuesto alto sin *soft-restart*.

Durante todas las ejecuciones de *irace*, los parámetros de tipo real se manejan con una precisión de dos decimales, manteniendo coherencia con la discretización definida en las

locaciones. Asimismo, los valores de calidad evaluados para cada configuración también se redondean a dos decimales, lo que permite estandarizar los criterios de evaluación y facilita la comparación entre soluciones. Finalmente, al utilizar las configuraciones por defecto de *irace*, se aplica el test de Friedman con un 95 % de confianza para comparar estadísticamente las configuraciones.

4.1.5. Metodología de validación de la solución

La validación se basa en aplicar el modelo STN-i sobre diversos escenarios definidos por combinaciones de algoritmos objetivo, conjuntos de instancias, configuraciones de hiperparámetros de *irace* y locaciones discretizadas. Para cada algoritmo se consideran uno o más conjuntos de instancias, evaluando cuatro configuraciones de *irace* que combinan dos niveles de presupuesto (bajo y alto) y la presencia o ausencia del mecanismo de *soft-restart*.

Cada tipo de escenario definido se ejecuta 20 veces por conjunto de instancias, utilizando semillas aleatorias fijas entre configuraciones con igual presupuesto. Esto permite realizar comparaciones controladas dentro del mismo conjunto de instancias y locación (por ejemplo, L_1), aislando así el efecto del uso de *soft-restart*. En el caso de PSO-X, se analizan por separado los subconjuntos de instancias *PSO-Mix*, *PSO-Mul* y *PSO-Uni*.

Por cada grupo de ejecuciones independientes de *irace* (20 repeticiones), se construyen tres modelos STN-i —uno por locación (L_1, L_2, L_3)— resultando en un total de 12 redes STN-i por conjunto de instancias (4 configuraciones $\times$ 3 locaciones). A partir de estas se derivan seis redes merged STN-i, cada una combinando dos configuraciones con igual presupuesto y diferente uso de *soft-restart*.

Cada red, ya sea STN-i o merged STN-i, se visualiza mediante el algoritmo de Fruchterman-Reingold [Fruchterman y Reingold, 1991], que simula un sistema de partículas para distribuir los nodos. Esta técnica mejora la legibilidad estructural y permite identificar patrones en la conectividad y convergencia de las trayectorias generadas por *irace*.

Para cada combinación de algoritmo y conjunto de instancias se realiza un análisis que considera conjuntamente las visualizaciones de las STN-i, las métricas asociadas a estas redes, las correspondientes merged STN-i con sus respectivas métricas, y un resumen complementario de datos derivados de las ejecuciones de *irace* (como uso de *soft-restart*, número de configuraciones evaluadas y estadísticas de iteración). Esta combinación de fuentes permite identificar patrones relevantes de exploración, convergencia y sensibilidad estructural frente al uso de *soft-restart*.

Finalmente, se presenta una discusión general integradora basada en los hallazgos obtenidos para cada algoritmo, destacando el valor descriptivo de las STNs-i como herramienta para comprender el comportamiento de *irace* bajo distintos escenarios y condiciones experimentales.

4.2. Análisis y resultados

Esta sección presenta y analiza los resultados obtenidos a partir de los escenarios experimentales y del procesamiento de las trayectorias generadas por *irace*, empleando los modelos STN-i y merged STN-i. El análisis se estructura según pares *algoritmo objetivo* – *conjunto de instancias*, lo que permite examinar en detalle el comportamiento del proceso de sintonización en función del tipo de problema abordado y del enfoque algorítmico utilizado. Los resultados se discuten a medida que se presentan, permitiendo una interpretación contextualizada de cada escenario.

En el caso de PSO-X, se reportan por separado las variantes *PSO-Mix*, *PSO-Mul* y *PSO-Uni*, considerando para cada una las locaciones discretizadas correspondientes (L_1 , L_2 , L_3). Para cada combinación se incluyen las redes generadas, sus visualizaciones y las tablas con métricas asociadas.

A partir de los 12 modelos STN-i iniciales por escenario (tres por locación y condición experimental), se generan nueve visualizaciones por escenario, agrupadas en tres variantes: (1) visualización completa, (2) sin nodos regulares de inicio, y (3) sin nodos regulares en general. Estas variantes permiten analizar los efectos estructurales de diferentes subconjuntos de configuraciones sobre la red.

En el caso de las redes merged STN-i, se siguen los mismos criterios. Para cada escenario, se generan tres modelos merged por locación, a partir de los modelos STN-i individuales, y se construyen nueve visualizaciones resultantes (tres por cada uno de los modelos merged). Esto permite examinar cómo varía la estructura global al consolidar múltiples trayectorias de ejecución bajo diferentes condiciones.

Las métricas consideradas incluyen número de nodos, proporciones por rol (regulares, élite, máximos), densidad, direccionalidad, conectividad, origen de configuraciones clave y accesibilidad hacia nodos de mayor calidad. En las redes merged STN-i se incorporan además métricas de solapamiento entre categorías. Todos los valores numéricos se reportan con una precisión de hasta cuatro decimales.

4.2.1. Resultados de MMASQAP (QAP-60)

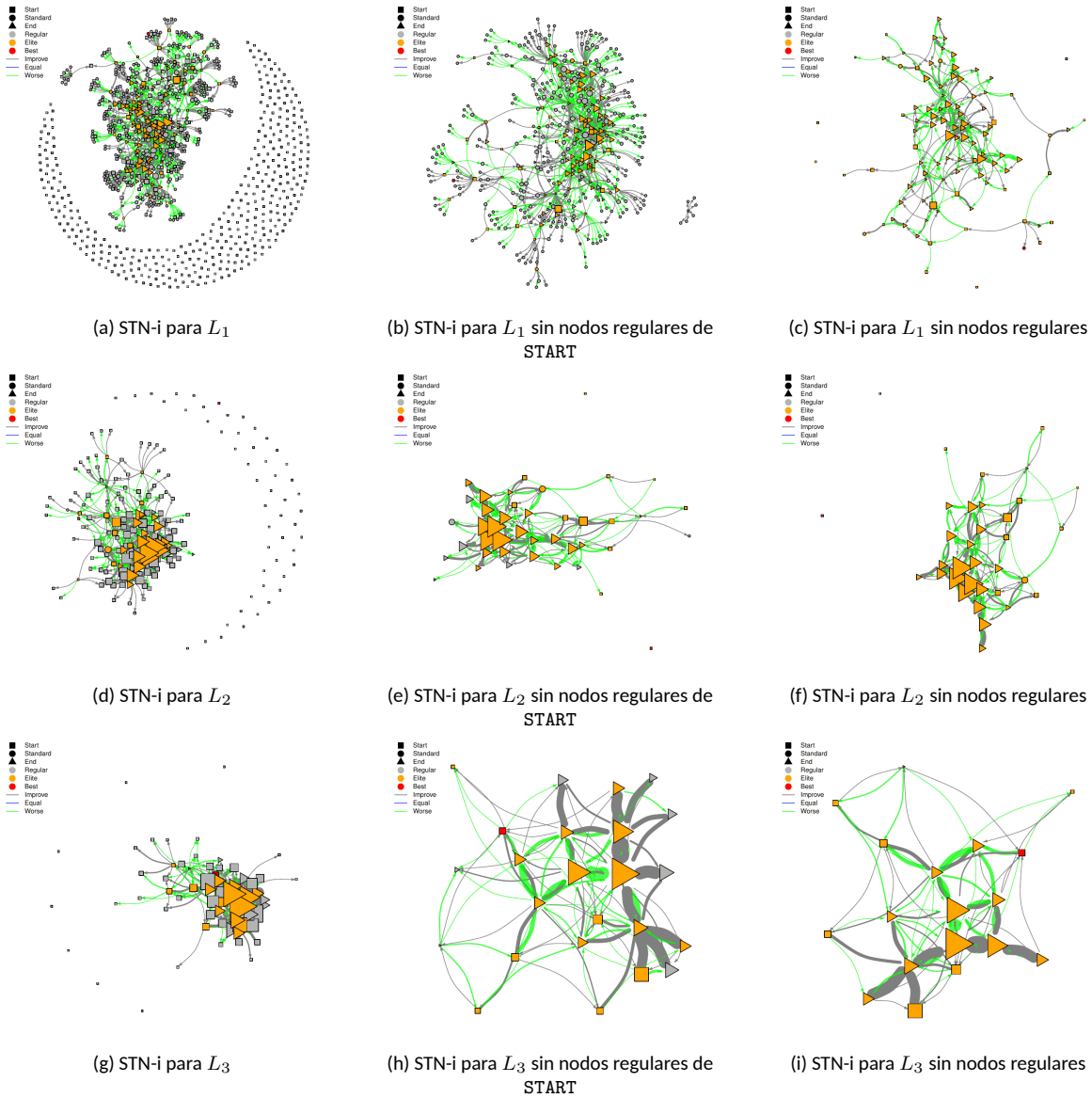


Figura 12: Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

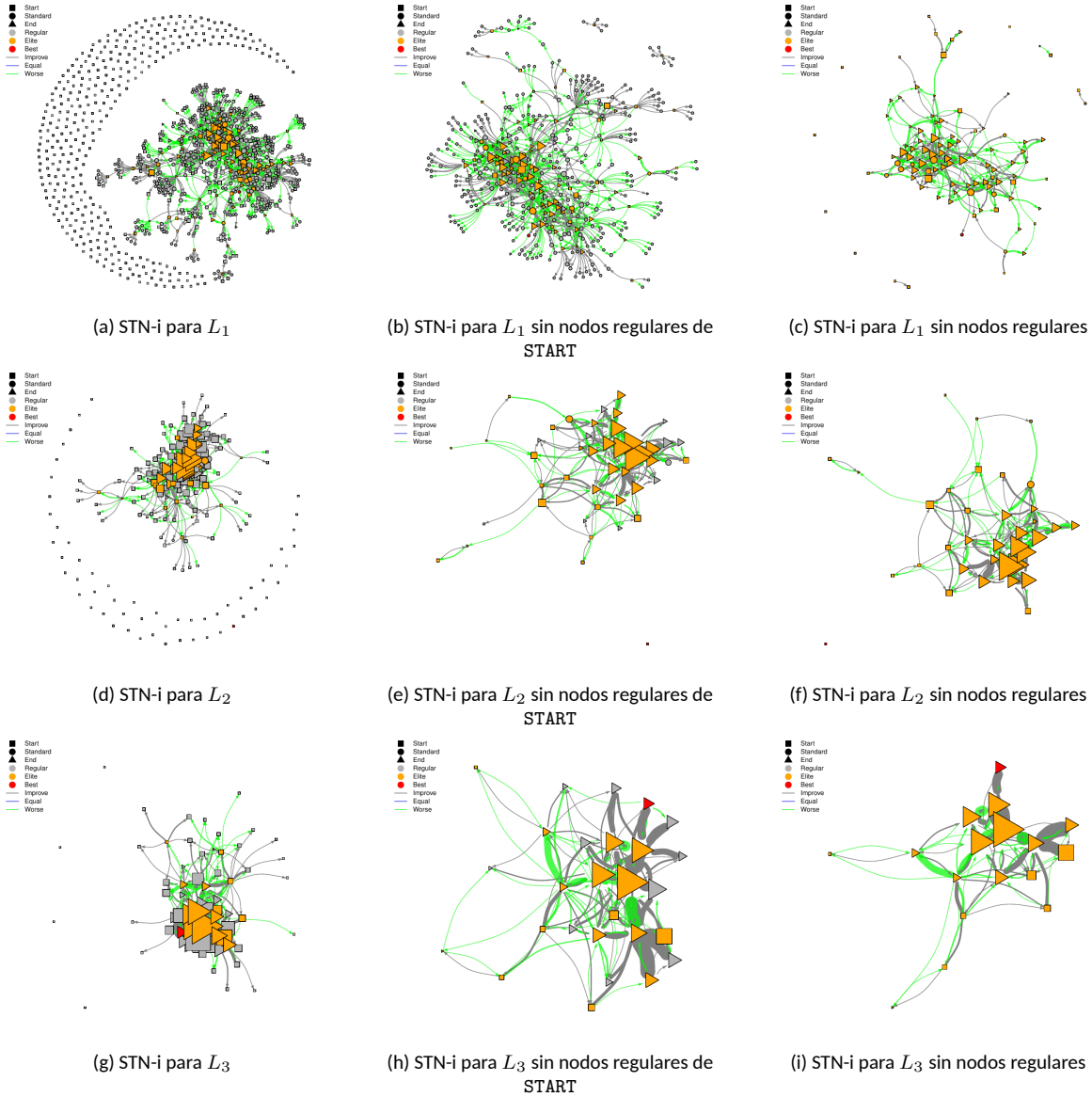


Figura 13: Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

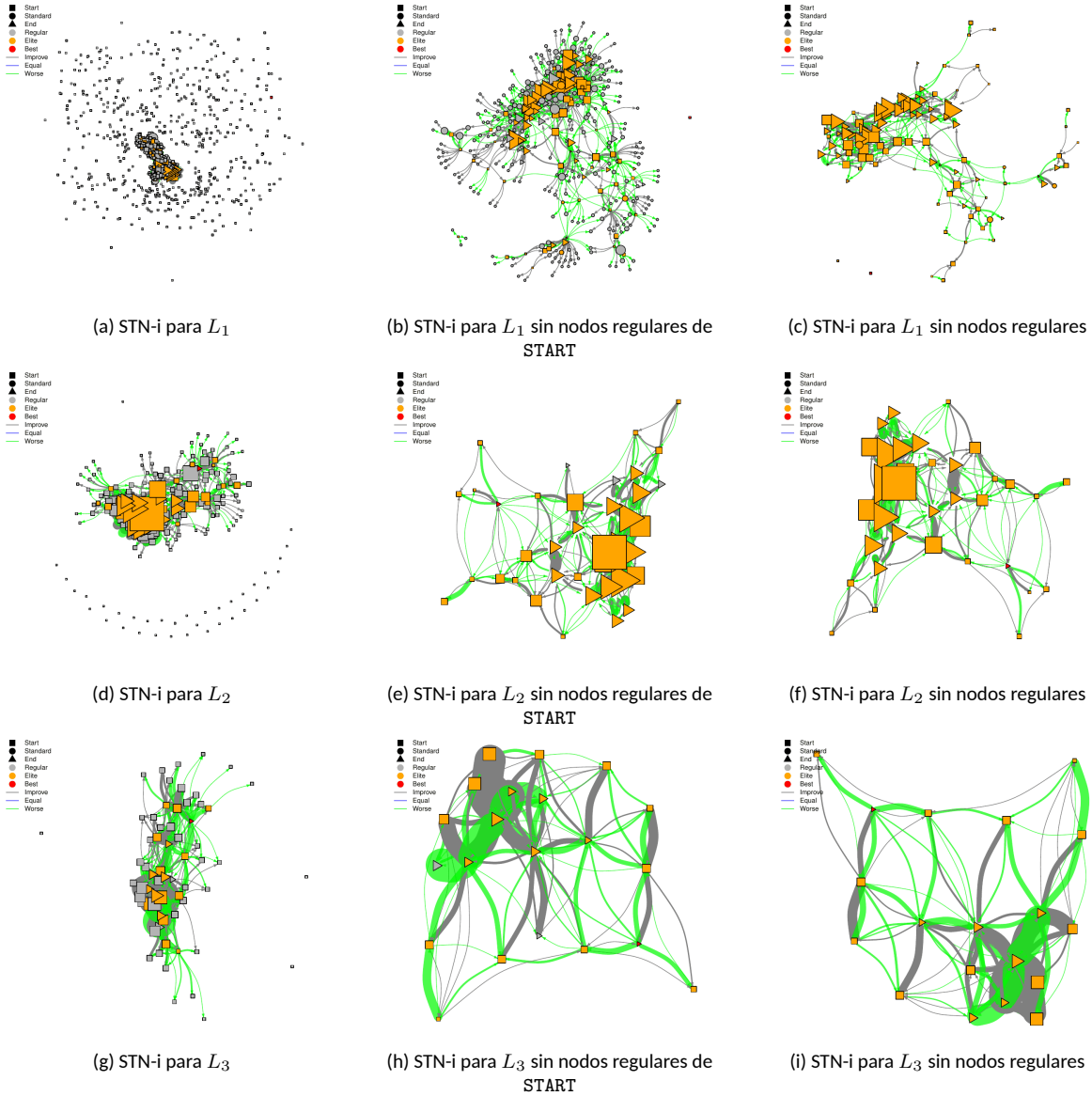


Figura 14: Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

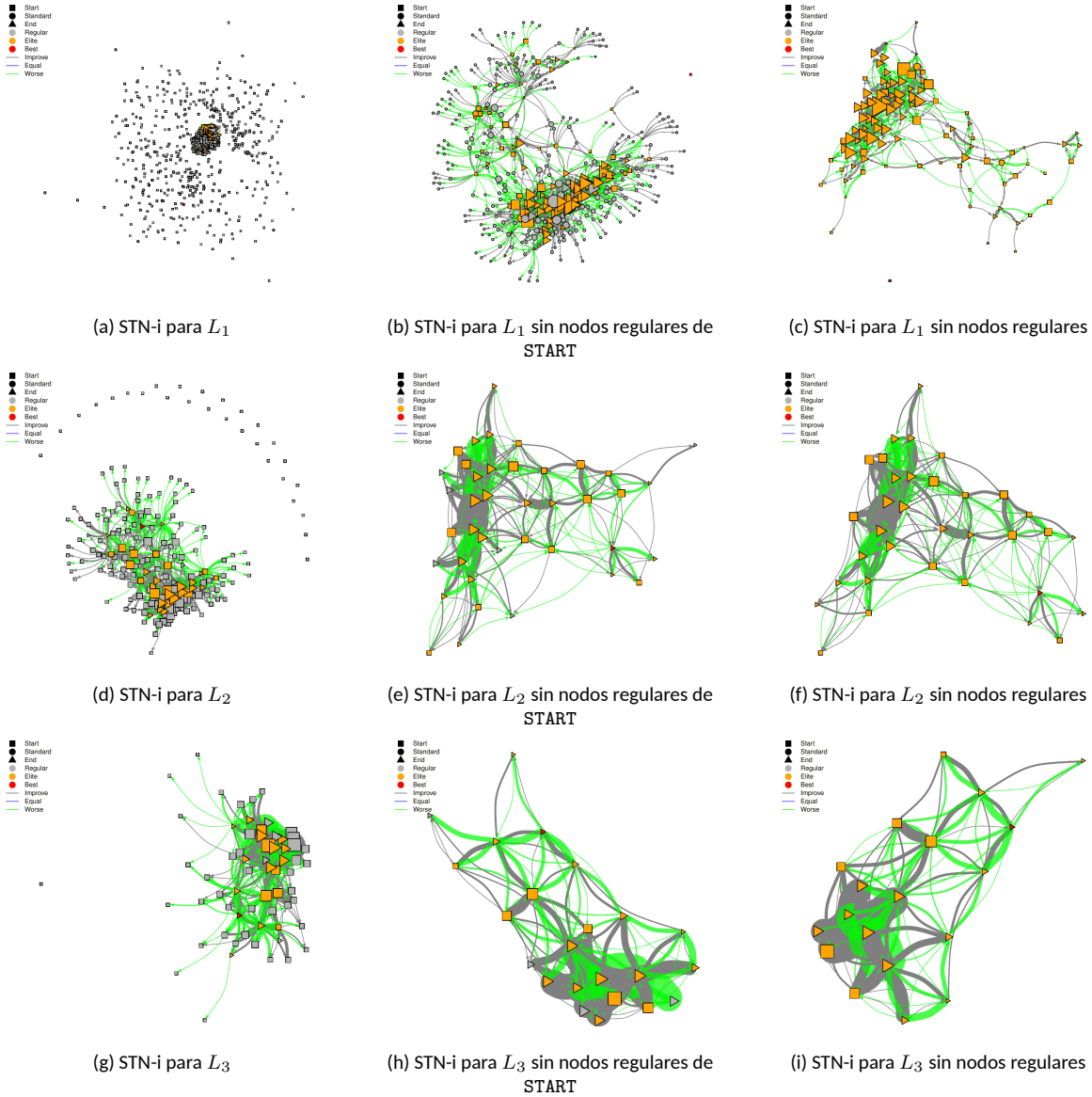


Figura 15: Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia QAP-60 utilizando el algoritmo MMASQAP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

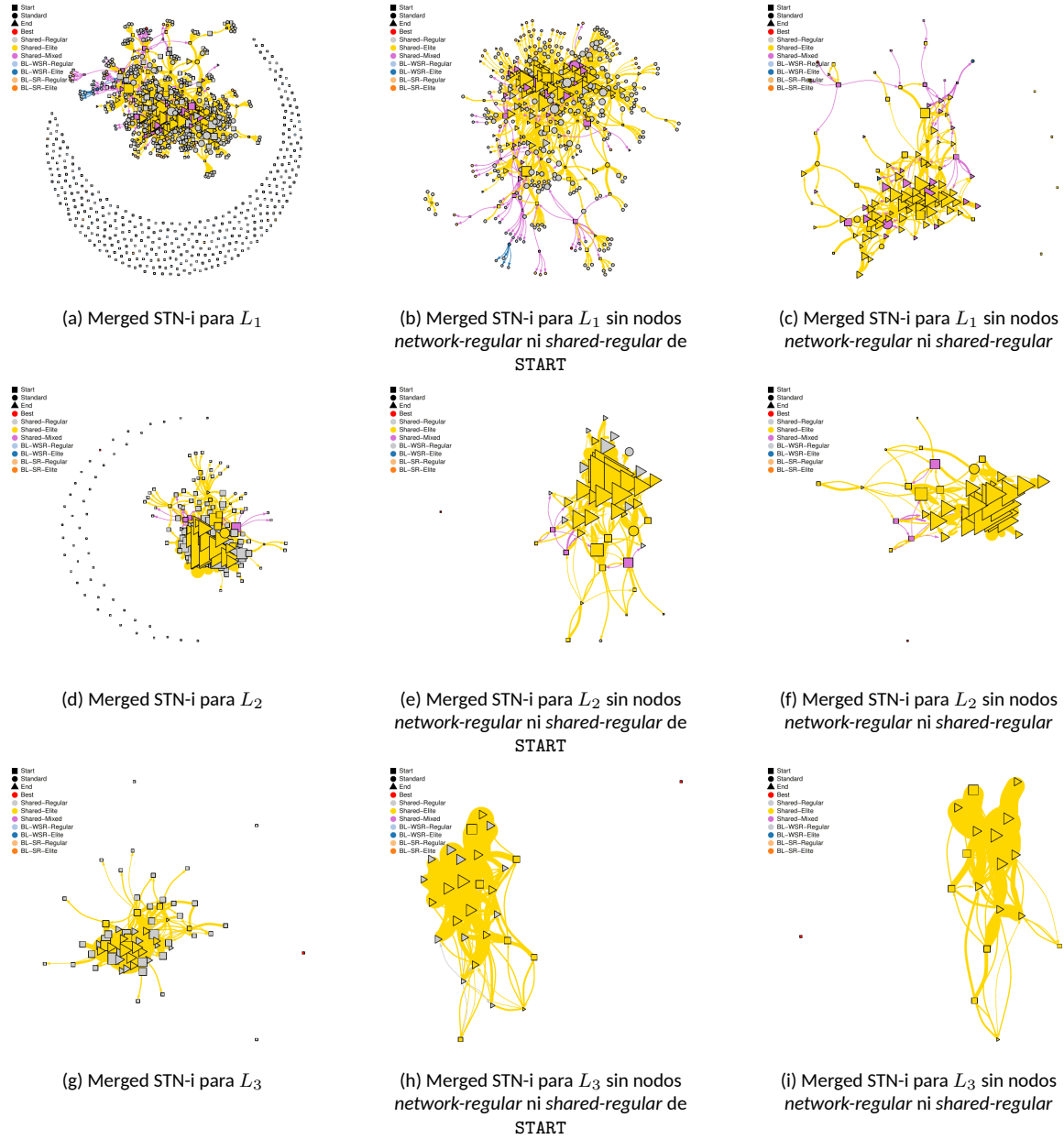


Figura 16: Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia QAP-60 con el algoritmo MMASQAP. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

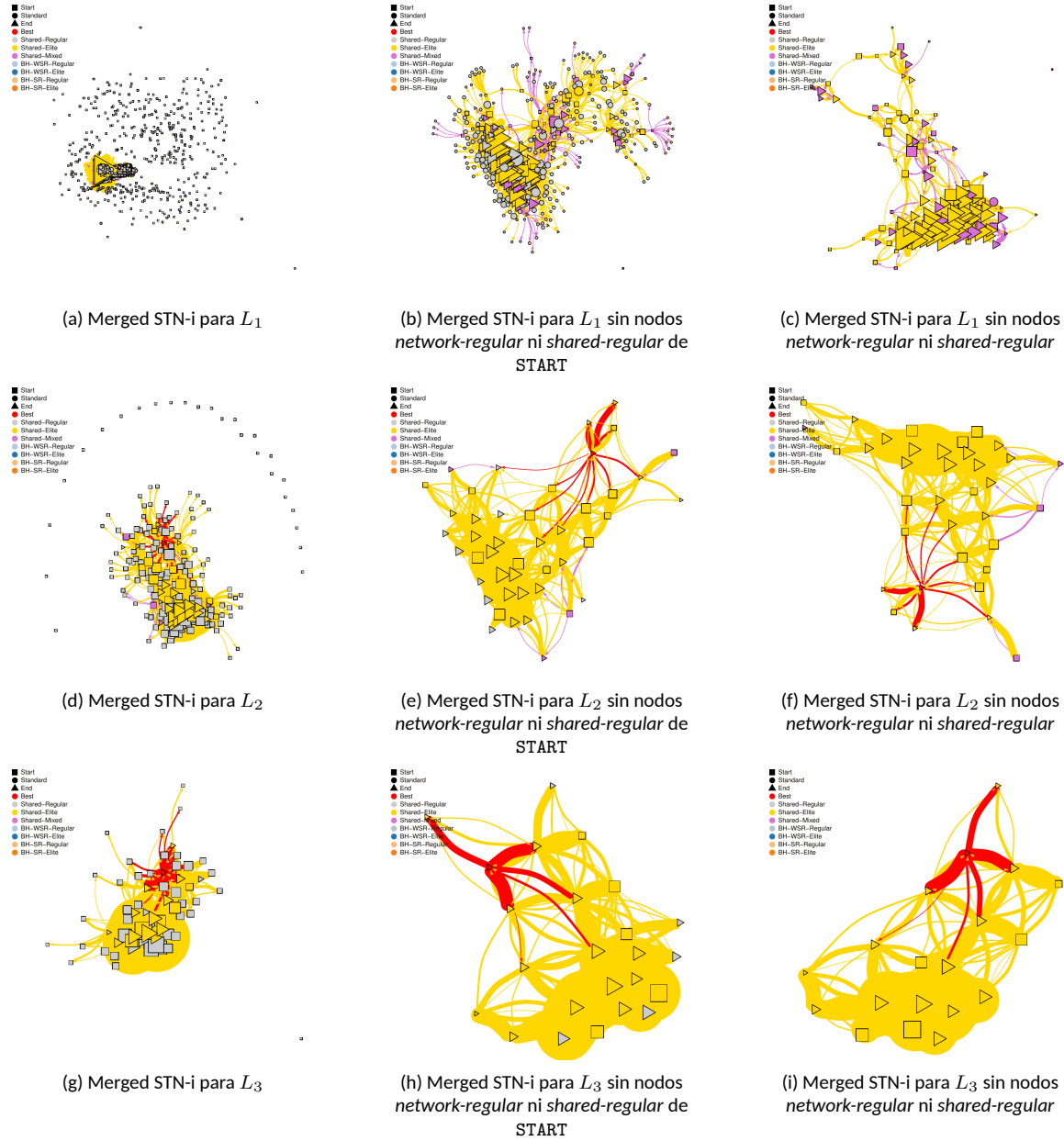


Figura 17: Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia QAP-60 con el algoritmo MMASQAP. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR
IRACE

Escenario	Locación	n	p_{nre}	p_{nel}	p_{nbt}	ed	$tend_{ed}$	p_{c-nre}	p_{c-nel}	$p_{c-n_{re}^{st}}$	$p_{c-n_{el}^{st}}$	$tend_c$	co	str_{nbt}^{in}	len_{pt}	pt
BL-WSR	L_1	938	0.8955	0.1034	0.0011	963	0.5327	0.8529	0.1471	0.9463	0.0537	0.2416	381	0.0500	1.5000	2
	L_2	194	0.8247	0.1701	0.0052	493	0.5720	0.8529	0.1471	0.9712	0.0288	0.2416	53	0.0000	0.0000	1
	L_3	64	0.7188	0.2656	0.0156	229	0.5546	0.8529	0.1471	0.9744	0.0256	0.2416	8	0.3500	2.2857	7
BL-SR	L_1	921	0.8936	0.1053	0.0011	1003	0.5304	0.8515	0.1485	0.9422	0.0578	0.2425	352	0.0500	5.4615	13
	L_2	194	0.8196	0.1753	0.0052	508	0.6339	0.8515	0.1485	0.9681	0.0319	0.2425	57	0.0000	0.0000	1
	L_3	64	0.7344	0.2500	0.0156	225	0.5956	0.8515	0.1485	0.9749	0.0251	0.2425	5	1.0000	4.0000	6
BH-WSR	L_1	1271	0.9213	0.0779	0.0008	1593	0.5185	0.9200	0.0800	0.9563	0.0437	0.1702	524	0.0000	0.0000	1
	L_2	196	0.8265	0.1684	0.0051	625	0.5936	0.9200	0.0800	0.9736	0.0264	0.1702	37	0.3500	3.2000	20
	L_3	64	0.7031	0.2813	0.0156	264	0.5417	0.9200	0.0800	0.9805	0.0195	0.1702	4	0.5000	4.5000	12
BH-SR	L_1	1305	0.9234	0.0759	0.0008	1992	0.4990	0.9155	0.0845	0.9722	0.0278	0.2119	491	0.0000	0.0000	1
	L_2	196	0.8265	0.1684	0.0051	720	0.5611	0.9155	0.0845	0.9792	0.0208	0.2119	29	0.6000	3.4667	15
	L_3	64	0.7031	0.2813	0.0156	296	0.5000	0.9155	0.0845	0.9883	0.0117	0.2119	2	0.7500	3.5000	6

Tabla 7: Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo MMASQP sobre la instancia QAP-60.

Escenarios	Locación	n	p_{nre}	p_{nel}	p_{nbt}	p_{nsh}	p_{nshre}	p_{nshel}	p_{nshmx}	ed	co	str_{nbt}^{in}	len_{pt}	pt
BL-WSR $\cup$ BL-SR	L_1	991	0.1211	0.0020	0.001	0.8759	0.7649	0.0827	0.0283	1247	363	0.0250	2.0000	2
	L_2	194	0.0000	0.0000	0.005	1.0000	0.8144	0.1649	0.0155	578	49	0.0000	0.0000	1
	L_3	64	0.0000	0.0000	0.016	1.0000	0.7188	0.2656	0.0000	244	5	0.0000	0.0000	1
BH-WSR $\cup$ BH-SR	L_1	1354	0.0975	0.0000	0.001	0.9025	0.8161	0.0613	0.0244	2518	458	0.0000	0.0000	1
	L_2	196	0.0000	0.0000	0.005	1.0000	0.8163	0.1582	0.0204	789	27	0.4750	5.1333	15
	L_3	64	0.0000	0.0000	0.016	1.0000	0.7031	0.2813	0.0000	309	2	0.6250	6.6000	5

Tabla 8: Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo MMASQP en la instancia QAP-60.

Escenario	Configuraciones	Iter. Mín	Iter. Máx	Iter. Prom	Uso de <i>soft-restart</i>
BL-WSR	2019	3	5	3.9500	–
BL-SR	2018	3	5	3.9500	0.1542
BH-WSR	4230	4	5	4.2500	–
BH-SR	4242	4	6	4.6500	0.3125

Tabla 9: Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia QAP-60 del algoritmo MMASQP.

4.2.2. Análisis de MMASQAP (QAP-60)

Dado que el conjunto de instancias QAP-60 presenta una alta homogeneidad, la variabilidad entre ejecuciones de *irace* se ve reducida considerablemente. Esto permite que el proceso de sintonización se mantenga más estable, ya que las configuraciones no se ven afectadas por diferencias estructurales significativas en el problema. En este contexto, al considerar la discretización de solo dos parámetros del algoritmo MMASQAP para construir las STN-i, se genera un espacio de configuraciones relativamente acotado y de baja complejidad. Esta condición favorece la visualización y la interpretación de las trayectorias exploradas, tanto en las redes STN-i como en su versión fusionada merged STN-i.

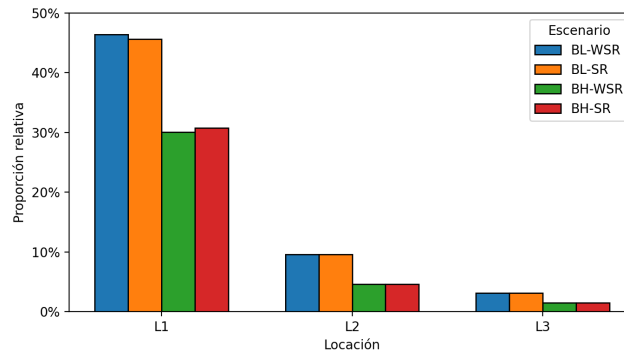


Figura 18: Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia QAP-60 del algoritmo MMASQAP.

Como se observa en la Figura 18, existe una tendencia consistente entre los distintos escenarios respecto a la proporción de locaciones (nodos) sobre el total de configuraciones generadas por *irace*. A pesar de que un mayor presupuesto computacional permite evaluar más configuraciones, las discretizaciones utilizadas (L_1 , L_2 y L_3) tienden a agrupar un mayor número de configuraciones en cada locación al reducir la granularidad. Esto refleja que las locaciones utilizadas logran condensar efectivamente el espacio de búsqueda, manteniendo una representación estructural coherente entre escenarios de diferente presupuesto.

Tal como se muestra en la Tabla 9, los escenarios evaluados presentan una cantidad mínima y máxima de iteraciones bastante estable, incluso al duplicar el presupuesto. Esto indica que el aumento de presupuesto no impacta directamente en el número de iteraciones, sino en la cantidad de configuraciones generadas por cada una de ellas. Este comportamiento es coherente con el funcionamiento interno de *irace*, que ajusta dinámicamente el tamaño de las muestras evaluadas por iteración en función del presupuesto disponible. Como consecuencia, se incrementa también la activación del mecanismo de *soft-restart* en escenarios con mayor presupuesto, lo cual se interpreta como una respuesta adaptativa ante una convergencia más rápida o prematura, producto de la mayor presión selectiva al evaluar más configuraciones por iteración.

El comportamiento estructural de las redes STN-i puede observarse en las Figuras 12–15 y

se respalda por las métricas de la Tabla 7. Visualmente, las redes construidas con locaciones más finas (L_1) presentan una gran cantidad de nodos desconexos, que corresponden principalmente a configuraciones descartadas tempranamente por *irace*. Esta fragmentación se reduce al utilizar discretizaciones más gruesas (L_2 y L_3), donde múltiples configuraciones se agrupan en regiones más amplias, generando nodos más densos y conectados.

Cuantitativamente, se observa que en todos los escenarios el número de arcos ed supera al número de nodos n , lo que indica una alta conectividad relativa. A su vez, la cantidad de componentes co disminuye al aumentar el tamaño de las locaciones, reflejando una menor fragmentación estructural. En cuanto a la métrica $tend_{ed}$, esta muestra una ligera inclinación hacia transiciones entre configuraciones con mejoras relativas de calidad. Esto sugiere que, dentro del subconjunto explorado por *irace*, las trayectorias predominantes tienden a mantener o mejorar el rendimiento, sin evidenciar un comportamiento fuertemente exploratorio o disruptivo.

La distribución de tipos de nodos en las STN-i muestra una clara predominancia de nodos compuestos por configuraciones regulares, lo cual se refleja en los altos valores de $p_{n_{re}}$. Esta proporción disminuye al aumentar el tamaño de las locaciones, ya que al agrupar más configuraciones en una misma región, aumenta la probabilidad de que alguna de ellas sea clasificada como élite. En consecuencia, las proporciones de nodos élite ($p_{n_{el}}$) y de nodos con mejores configuraciones ($p_{n_{bt}}$) aumentan, aunque esto se debe a la reducción del total de nodos, y no a un incremento en el desempeño general.

Este patrón también se refleja en la composición interna de los nodos. Más del 85 % de las configuraciones corresponden a configuraciones descartadas ($p_{c-n_{re}}$), mientras que las élite ($p_{c-n_{el}}$) constituyen una fracción mucho menor. Esta diferencia se acentúa en los nodos iniciales del proceso de sintonización, donde los valores de $p_{c-n_{re}^{st}}$ superan ampliamente a los de $p_{c-n_{el}^{st}}$, evidenciando una fuerte presión selectiva desde las primeras iteraciones. Las configuraciones élite representan menos del 10 % al inicio, lo cual es coherente con la mecánica interna de *irace* y refuerza la interpretación de los nodos regulares iniciales como elementos descartados estructuralmente relevantes.

En cuanto a los nodos identificados como mejores, la métrica $str_{n_{bt}}^{in}$ muestra variaciones relevantes entre escenarios. Algunos presentan baja conectividad y fuerza de entrada, mientras que en otros, como L_3 del escenario BL-SR, se alcanza el valor máximo, indicando que dicho nodo concentra una alta proporción de trayectorias entrantes. Las métricas len_{pt} y pt complementan este análisis: en ciertos casos, los mejores nodos coinciden con nodos de inicio (trayectorias de longitud cero), lo cual puede deberse tanto a su selección temprana como a la sensibilidad de la medición de calidad basada en la media, susceptible a valores atípicos. En contraste, otros escenarios presentan múltiples trayectorias de longitud promedio entre 1.5 y 5.46, lo que refleja una red más interconectada y con diversas rutas hacia configuraciones de alto rendimiento.

Finalmente, la métrica $tend_c$ indica que solo una pequeña proporción de configuraciones es seleccionada como élite en más de una iteración, con valores que oscilan entre un 17 % y un

24 %. Esta baja tasa de supervivencia es coherente con la naturaleza altamente selectiva de *irace*. Aunque este valor es superior al observado en la primera iteración, puede explicarse porque, en etapas posteriores, la cantidad de configuraciones evaluadas por iteración disminuye, aumentando la probabilidad relativa de que una configuración permanezca activa y a su vez indicando un mayor nivel de explotación sobre el conjunto restante. En conjunto, estos resultados evidencian la intensidad del proceso de descarte y resaltan el valor de las STN-i como herramienta para analizar dicha dinámica evolutiva.

Las redes merged STN-i construidas a partir de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, presentan una estructura más consolidada que las STN-i individuales. Las visualizaciones (Figuras 16 y 17) muestran una alta presencia de nodos compartidos, donde destacan los nodos élite compartidos (en amarillo), junto a una considerable proporción de nodos regulares compartidos (en gris), incluyendo algunos nodos de inicio. Al igual que en las redes individuales, la granularidad de las locaciones afecta la cantidad de nodos aislados, los cuales se reducen progresivamente de L_1 a L_3 .

Desde el punto de vista estructural, la Tabla 8 confirma estas observaciones. En L_1 se mantiene una alta fragmentación, mientras que en L_2 y L_3 las redes presentan menos componentes (co) y una alta cantidad de arcos (ed), lo que evidencia una conectividad elevada en las regiones activas de la red fusionada. En las vistas filtradas se observa una estructura centrada en ciertos nodos élite, visualmente resaltados por su tamaño, que actúan como puntos de atracción.

Uno de los aspectos más relevantes de las redes merged STN-i es la proporción de nodos compartidos ($p_{n_{sh}}$), que es muy elevada en todos los escenarios y alcanza el 100 % en las locaciones L_2 y L_3 . Esto indica una convergencia estructural significativa entre ejecuciones independientes de *irace*. La mayoría de estos nodos compartidos son regulares ($p_{n_{shre}}$), lo que sugiere que ciertas regiones descartadas tienden a ser recorridas sistemáticamente. Por otra parte, los nodos élite compartidos ($p_{n_{shel}}$), aunque menos frecuentes, reflejan regiones de calidad recurrente. La baja proporción de nodos mixtos compartidos ($p_{n_{shmx}}$) indica una baja ambigüedad entre ejecuciones respecto a la evaluación estructural de las locaciones compartidas.

La presencia de nodos asociados a las mejores configuraciones ($p_{n_{bt}}$) es marginal, al igual que en las redes individuales. Sin embargo, las métricas estructurales muestran diferencias notables según el presupuesto. En los escenarios de menor presupuesto, algunos nodos mejores coinciden con nodos de inicio y tienen fuerza de entrada nula ($str_{n_{bt}}^{in} = 0$), generando trayectorias triviales ($len_{pt} = 0, pt = 1$). En contraste, en escenarios con mayor presupuesto, se observan mejores nodos con alta conectividad entrante y múltiples trayectorias hacia ellos, especialmente en locaciones L_2 y L_3 de la fusión $BH-WSR \cup BH-SR$. Esto sugiere una estructura más accesible hacia regiones prometedoras, con trayectorias variadas que permiten alcanzar configuraciones de alta calidad dentro del espacio explorado.

4.2.3. Resultados de ACOTSP (TSP-2000)

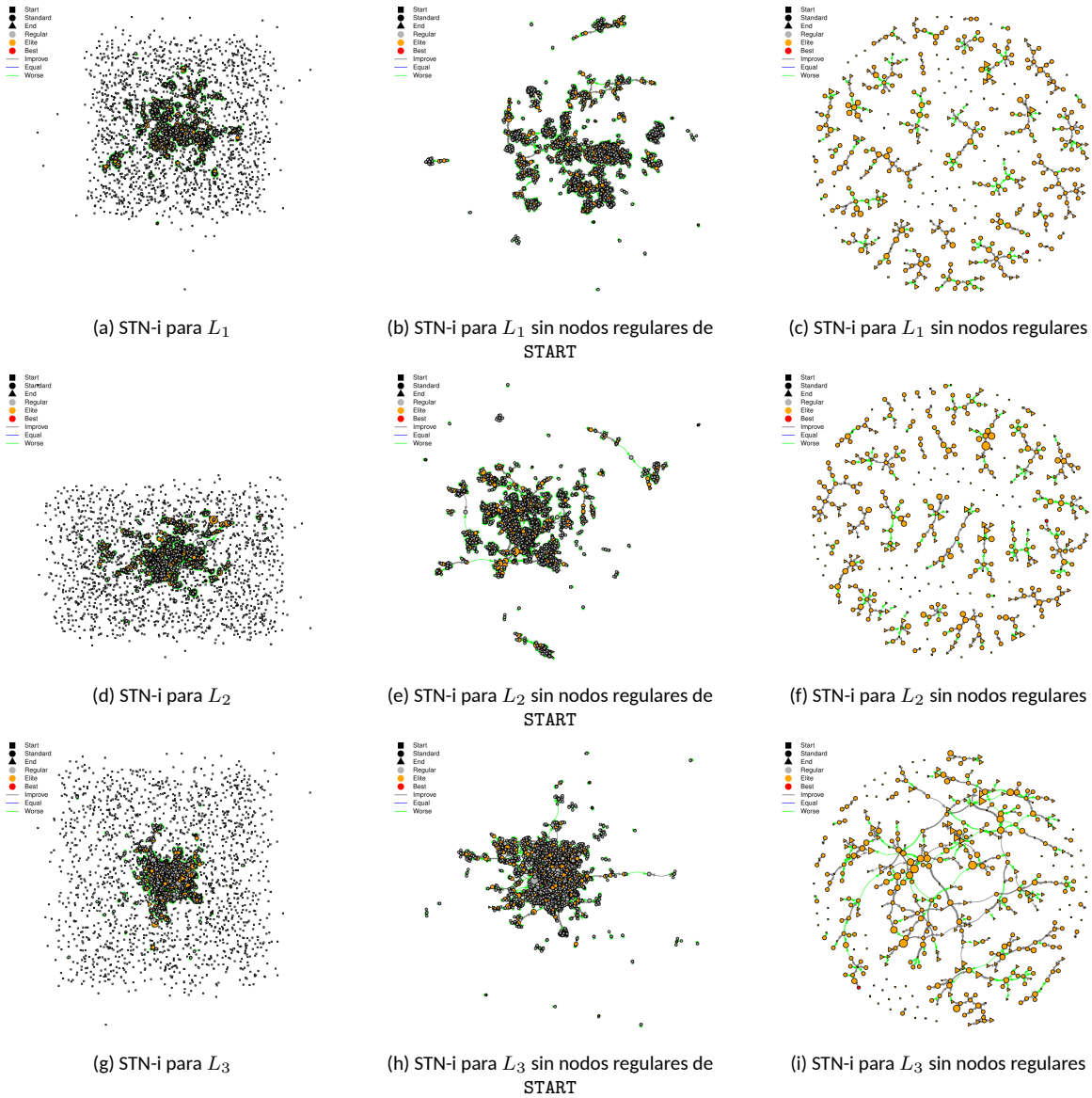


Figura 19: Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

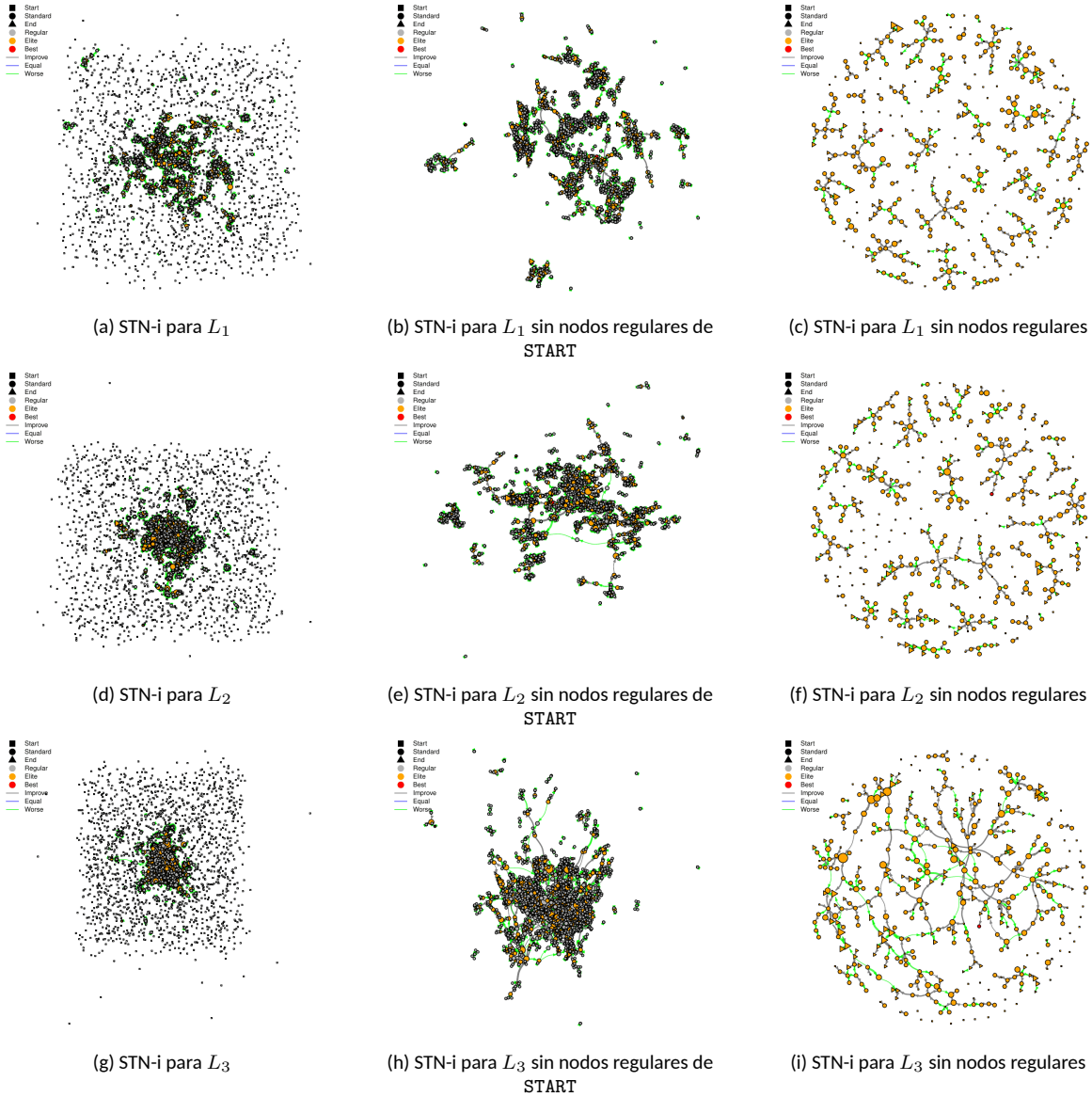


Figura 20: Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

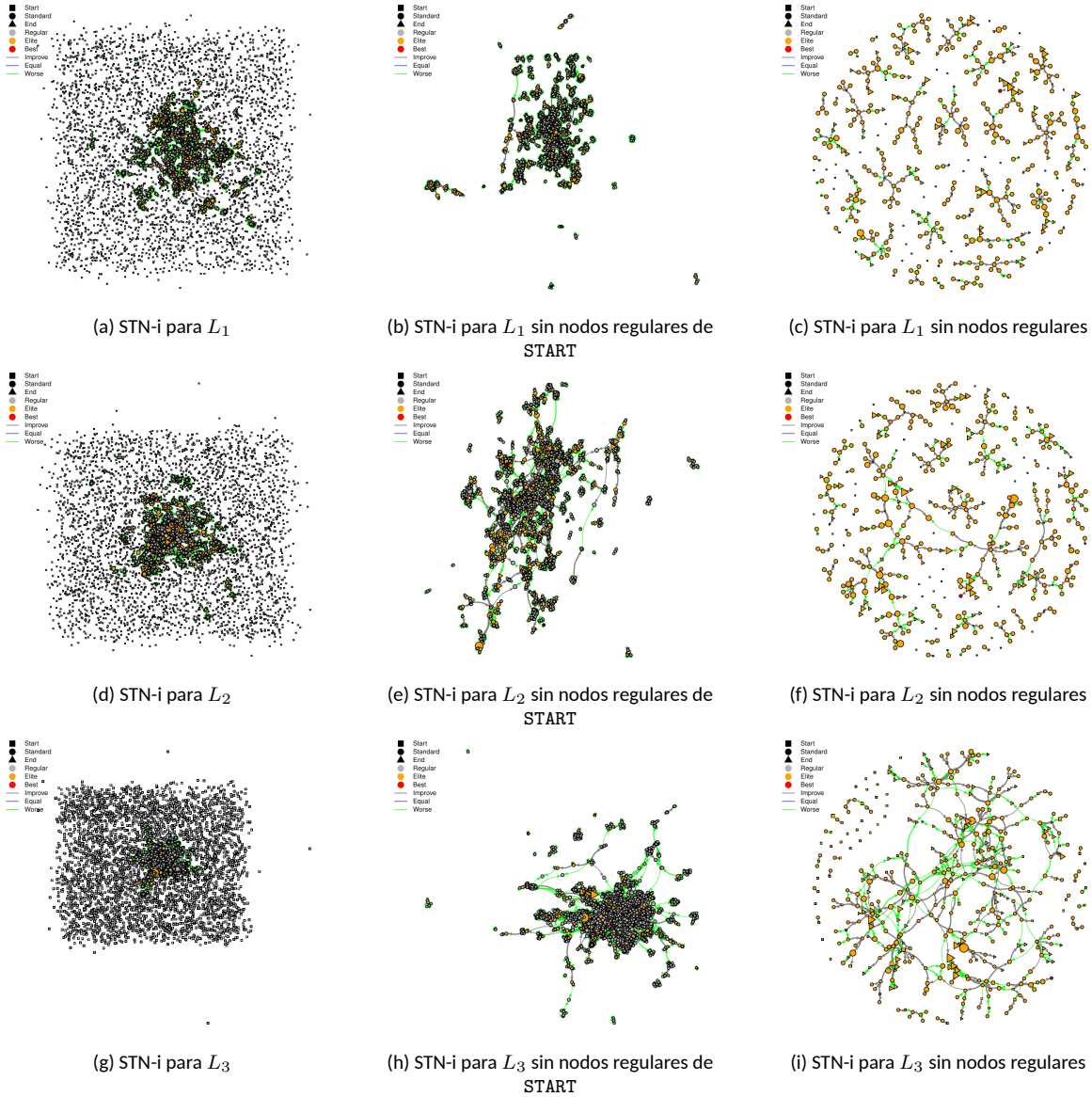


Figura 21: Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

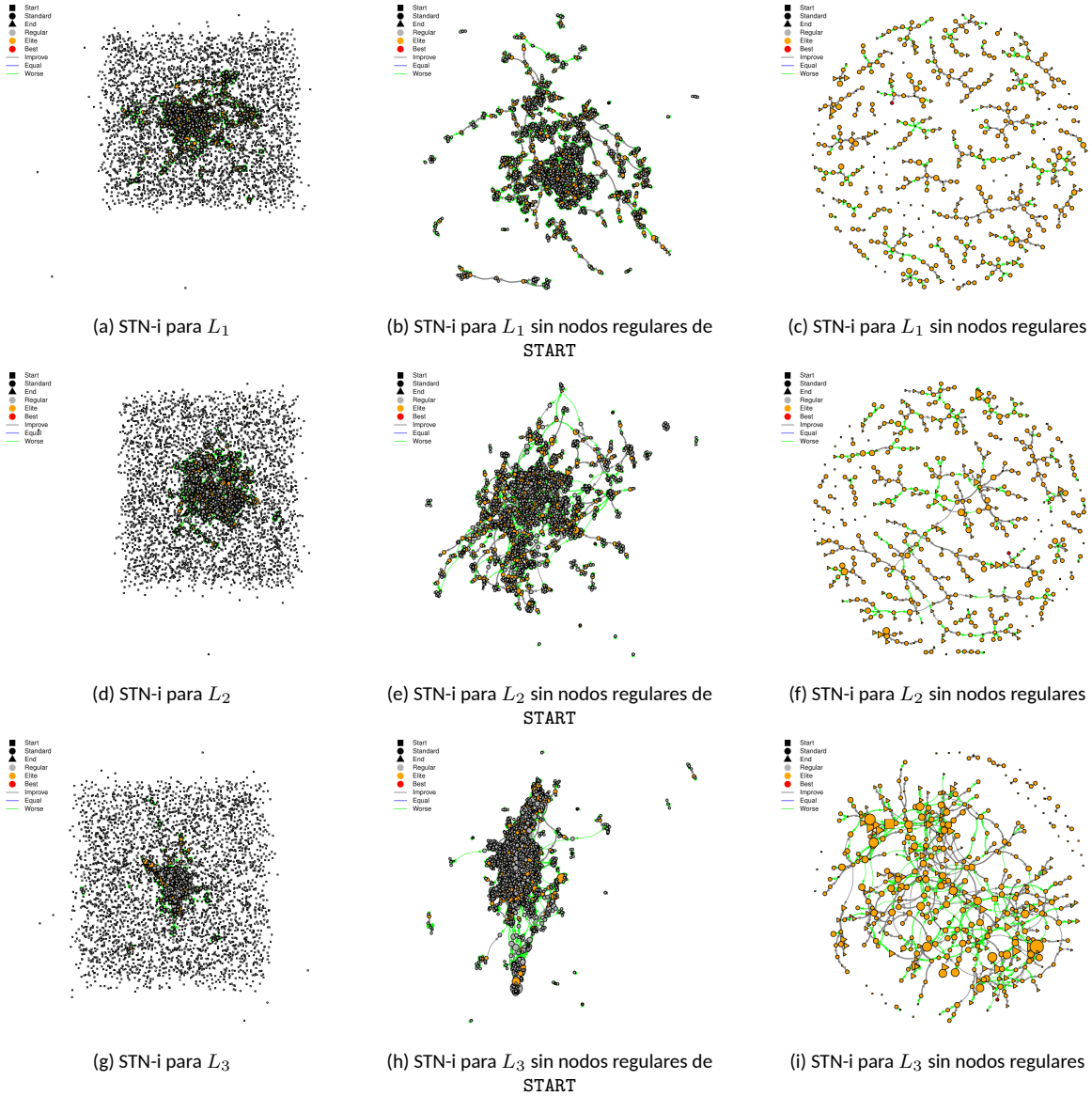


Figura 22: Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia TSP-2000 utilizando el algoritmo ACOTSP. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

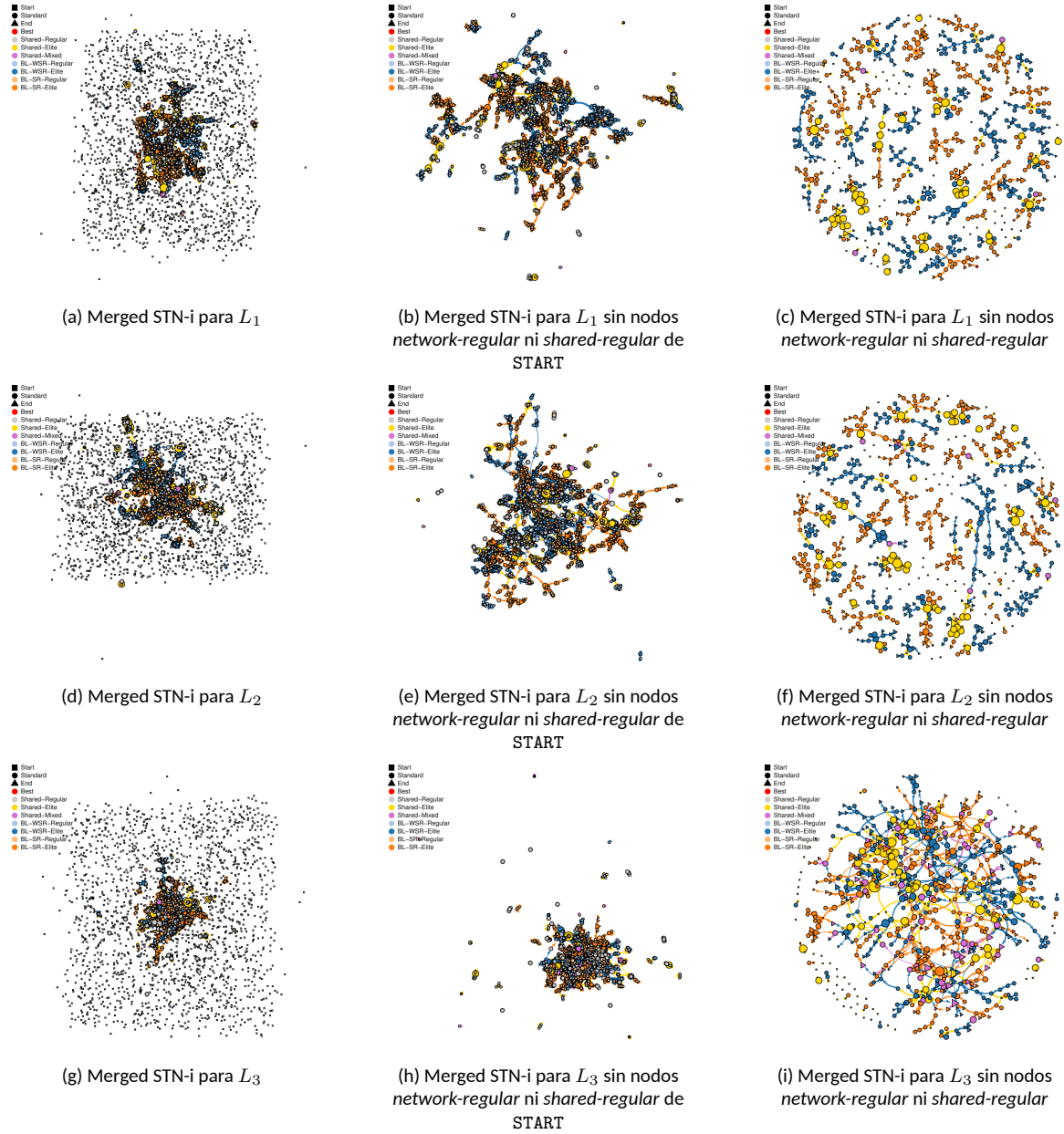


Figura 23: Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia TSP-2000 con el algoritmo ACOTSP. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos network-regular ni shared-regular provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

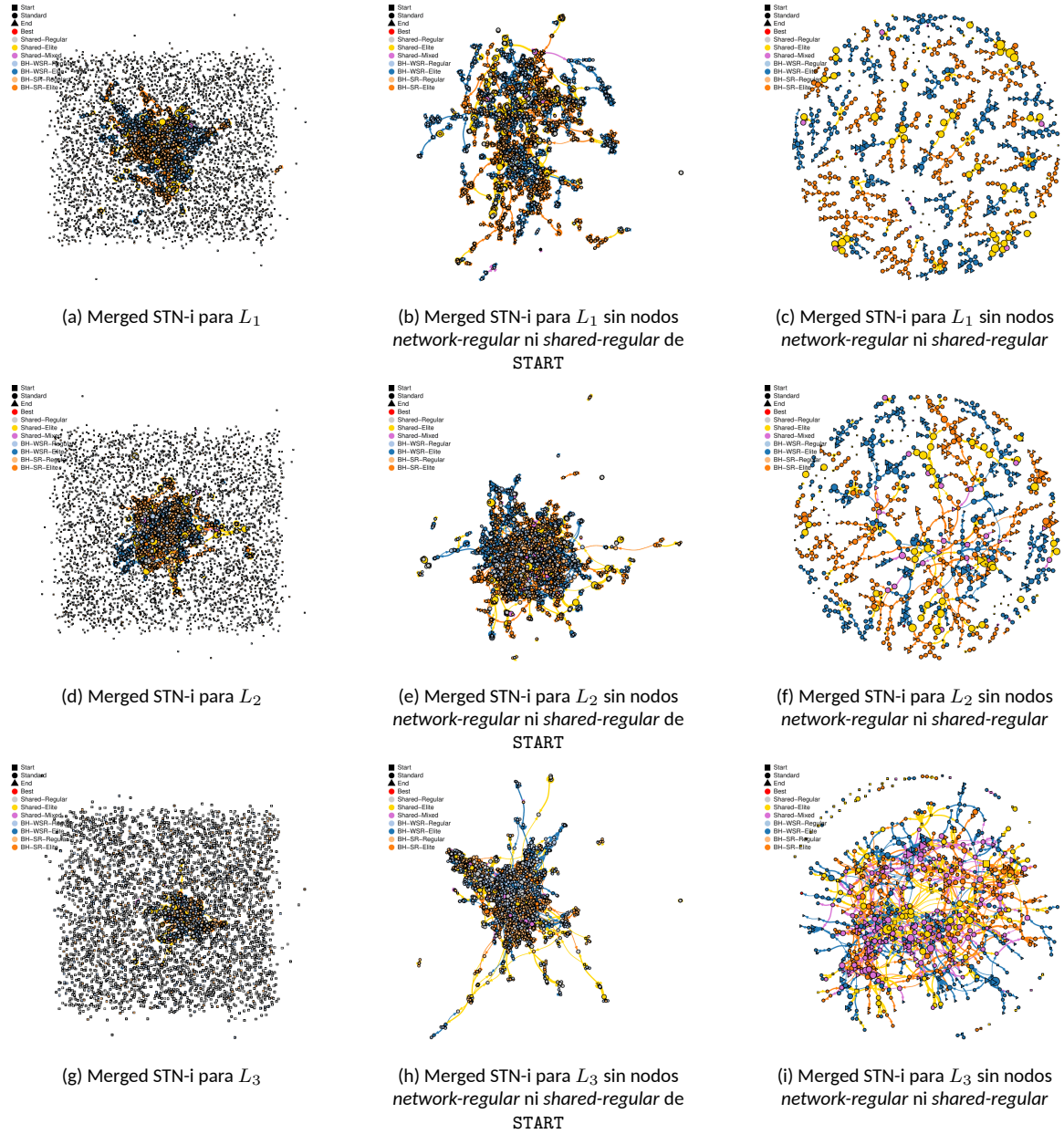


Figura 24: Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia *TSP-2000* con el algoritmo ACOTSP. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

**DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR
IRACE**

Escenario	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	ed	$tend_{ed}$	$p_{c-n_{re}}$	$p_{c-n_{el}}$	$p_{c-n_{st}^e}$	$p_{c-n_{st}^d}$	$tend_c$	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR	L_1	8790	0.9512	0.0487	0.0001	6970	0.1638	0.8870	0.1130	0.8972	0.1028	0.2444	1820	0.0500	10.0000	1
	L_2	8755	0.9511	0.0488	0.0001	6952	0.1638	0.8870	0.1130	0.8973	0.1027	0.2444	1809	0.0500	10.0000	1
	L_3	8426	0.9497	0.0502	0.0001	6863	0.1607	0.8870	0.1130	0.8975	0.1025	0.2444	1780	0.0500	6.0000	1
BL-SR	L_1	8796	0.9503	0.0496	0.0001	6976	0.1492	0.8798	0.1202	0.8973	0.1027	0.2851	1820	0.0500	12.0000	1
	L_2	8766	0.9501	0.0497	0.0001	6959	0.1502	0.8798	0.1202	0.8973	0.1027	0.2851	1812	0.0500	12.0000	1
	L_3	8438	0.9493	0.0506	0.0001	6876	0.1467	0.8798	0.1202	0.8977	0.1023	0.2851	1777	0.0500	12.0000	1
BH-WSR	L_1	17164	0.9737	0.0263	0.0001	13504	0.1846	0.9323	0.0677	0.9492	0.0508	0.2107	3660	0.0500	18.0000	1
	L_2	16912	0.9735	0.0265	0.0001	13386	0.1842	0.9323	0.0677	0.9492	0.0508	0.2107	3638	0.0500	18.0000	1
	L_3	15370	0.9717	0.0282	0.0001	12733	0.1697	0.9323	0.0677	0.9483	0.0517	0.2107	3597	0.0500	8.0000	1
BH-SR	L_1	17109	0.9731	0.0268	0.0001	13451	0.1494	0.9342	0.0658	0.9541	0.0459	0.1885	3659	0.0500	5.0000	1
	L_2	16815	0.9728	0.0271	0.0001	13322	0.1482	0.9342	0.0658	0.9541	0.0459	0.1885	3629	0.0500	5.0000	1
	L_3	15258	0.9709	0.0290	0.0001	12781	0.1505	0.9342	0.0658	0.9557	0.0443	0.1885	3588	0.0500	21.1000	10

Tabla 10: Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo ACOTSP sobre la instancia TSP-2000.

Escenarios	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	$p_{n_{sh}}$	$p_{n_{shre}}$	$p_{n_{shel}}$	$p_{n_{shmx}}$	ed	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR $\cup$ BL-SR	L_1	14592	0.7525	0.0423	0.000	0.2052	0.1964	0.0082	0.0006	12590	2002	0.0250	15.0000	1
	L_2	14499	0.7494	0.0421	0.000	0.2084	0.1993	0.0083	0.0008	12546	1972	0.0250	15.0000	1
	L_3	13532	0.7141	0.0396	0.000	0.2462	0.2320	0.0090	0.0052	12351	1947	0.0250	15.0000	1
BH-WSR $\cup$ BH-SR	L_1	29213	0.8028	0.0240	0.000	0.1732	0.1694	0.0034	0.0004	25189	4025	0.0250	20.0000	1
	L_2	28524	0.7940	0.0236	0.000	0.1824	0.1778	0.0035	0.0011	24940	3981	0.0250	20.0000	1
	L_3	24388	0.7240	0.0201	0.000	0.2559	0.2444	0.0043	0.0071	23720	3919	0.0250	22.4118	34

Tabla 11: Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo ACOTSP en la instancia TSP-2000.

Escenario	Configuraciones	Iter. Mín	Iter. Máx	Iter. Prom	Uso de <i>soft-restart</i>
BL-WSR	8790	7	9	8.0000	–
BL-SR	8796	7	11	8.6500	0.0000
BH-WSR	17166	8	11	9.0000	–
BH-SR	17112	8	9	8.6000	0.0000

Tabla 12: Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia TSP-2000 del algoritmo ACOTSP.

4.2.4. Análisis de ACOTSP (TSP-2000)

El conjunto *TSP-2000* se caracteriza por su alta homogeneidad, lo cual, al igual que en el caso de *MMA SQAP*, favorece una ejecución estable del proceso de sintonización mediante *irace*. No obstante, el espacio de búsqueda asociado a ACOTSP presenta una complejidad significativamente superior debido a la gran cantidad de parámetros involucrados (11 en total), de los cuales 8 son discretizados para la construcción de las STN-i.

Entre estos parámetros, destacan aquellos que definen el tipo de algoritmo base (5 posibles valores) y el uso de búsqueda local (4 valores posibles), ambos con fuerte impacto en la dinámica exploratoria. Esta complejidad se ve amplificada por la existencia de 5 parámetros condicionales (4 discretizados), cuya activación depende del valor de otros parámetros, lo que puede reducir la efectividad de ciertas discretizaciones al producir configuraciones que terminan siendo parcialmente inactivas.

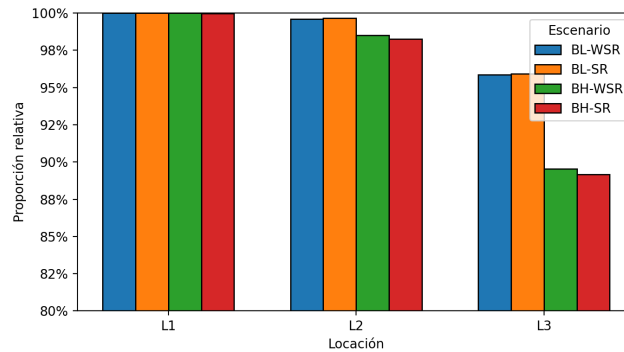


Figura 25: Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia *TSP-2000* del algoritmo ACOTSP.

La Figura 25 revela que la capacidad de reducción estructural de las locaciones es limitada, especialmente en L_1 y L_2 , donde más del 95 % de las configuraciones generadas corresponden a nodos únicos. Incluso en L_3 , donde se aplica una granularidad más gruesa, la reducción sigue siendo moderada: en torno al 96 % para escenarios de bajo presupuesto y apenas inferior al 90 % para escenarios de alto presupuesto. Este comportamiento sugiere que, debido a la complejidad del espacio combinatorio de ACOTSP, las configuraciones tienden a dispersarse ampliamente, dificultando su agrupamiento incluso con discretizaciones amplias.

Según la Tabla 12, el número de configuraciones generadas es coherente con el presupuesto asignado: alrededor de 8.700 para bajo presupuesto y más de 17.000 para alto. A pesar de esta diferencia, el número de iteraciones se mantiene estable entre 8 y 9 en promedio. Esto concuerda con la lógica operativa de *irace*, donde un mayor presupuesto se traduce en un mayor número de configuraciones evaluadas por iteración, sin alterar significativamente la cantidad de iteraciones.

El mecanismo de *soft-restart* se encuentra habilitado en algunos escenarios, pero su uso

efectivo es nulo. Esta inactividad puede explicarse por la alta complejidad estructural del espacio de búsqueda, donde la variabilidad introducida por los parámetros y sus interacciones condicionales impide que se generen patrones repetitivos o estancamientos, condiciones típicas para activar este mecanismo. La riqueza combinatoria inherente a ACOTSP promueve una dispersión que preserva la diversidad entre iteraciones, haciendo innecesario el reinicio o la re-exploración prematura.

Las redes STN-i visualizadas en las Figuras 19 a 22 presentan una alta proporción de nodos regulares generados en la primera iteración que fueron descartados tempranamente, patrón visible en todos los escenarios y locaciones. Esta acumulación se explica por la alta complejidad del espacio de búsqueda de ACOTSP y por el hecho de que cada red STN-i se construye a partir de 20 ejecuciones independientes por escenario, lo que naturalmente incrementa la cantidad de nodos desconectados, especialmente aquellos correspondientes a configuraciones descartadas tempranamente.

Una vez excluidos los nodos regulares iniciales (subfiguras (b), (e), (h)), se evidencian estructuras más densas aunque aún dominadas por nodos regulares. Las subfiguras (c), (f), (i), que muestran solo nodos élite, presentan redes con múltiples componentes desconectadas, revelando que las configuraciones de alta calidad no forman agrupamientos estructurados. No obstante, en escenarios de alto presupuesto con locación L_3 , se observa una mejora parcial en la conectividad entre estos nodos élite.

En términos cuantitativos (Tabla 10), el número de nodos (n) es consistentemente mayor al de arcos (ed), indicando una estructura más dispersa que en otros algoritmos. Las componentes conexas (co) se mantienen relativamente estables entre locaciones, sugiriendo que la granularidad de discretización no impacta significativamente la fragmentación de la red. Además, la métrica $tend_{ed}$ varía entre 0.14 y 0.18, lo cual evidencia una débil direccionalidad positiva: la mayoría de las transiciones entre nodos se asocian a un empeoramiento relativo en el rendimiento, dificultando la identificación de trayectorias estructuradas de mejora progresiva.

En todos los escenarios, las proporciones de nodos regulares ($p_{n_{re}}$) oscilan entre 94 % y 97 %, reflejando una estructura dominada por configuraciones descartadas. Los nodos élite ($p_{n_{el}}$) y los nodos mejores ($p_{n_{bt}}$) representan una fracción mínima de la red, con valores que rara vez superan el 5 % y 0.01 %, respectivamente. Esta baja densidad no implica falta de calidad, sino una distribución ampliamente dispersa que diluye su representación estructural.

En cuanto a la composición interna de los nodos, la proporción de configuraciones regulares ($p_{c-n_{re}}$) varía desde 87–88 % en escenarios de bajo presupuesto hasta 93 % en los de alto, evidenciando una mayor exploración pero también un menor filtrado selectivo. Las métricas de inicio ($p_{c-n_{re}^{st}}$, $p_{c-n_{el}^{st}}$) muestran comportamientos similares, con variaciones marginales de 1 % respecto al total, lo cual indica que el patrón de descarte es consistente desde las primeras iteraciones.

Los nodos asociados a las mejores configuraciones presentan una baja fuerza de entrada

($str_{nbt}^{in} = 0.05$), lo que indica que no actúan como polos estructurales dentro de la red. La cantidad de caminos cortos hacia estos nodos es generalmente uno ($pt = 1$), salvo en el escenario BH-SR con L_3 , donde existen 10 trayectorias con una longitud promedio de 21.1. Esta dispersión de trayectorias sugiere que los mejores nodos no coinciden con configuraciones iniciales, sino que son alcanzados a través de exploraciones más profundas.

La métrica $tend_c$ —proporción de configuraciones élite repetidas en más de una iteración— es mayor en escenarios de bajo presupuesto (0.24–0.28) que en los de alto (0.18–0.21). Esto puede explicarse porque un espacio más limitado genera mayor presión selectiva, promoviendo la retención de ciertas configuraciones. En cambio, al incrementarse el presupuesto, la diversidad también lo hace, reduciendo la probabilidad de recurrencia de configuraciones élite.

Las Figuras 23 y 24 muestran que las redes fusionadas mantienen una baja proporción de nodos compartidos, visualmente reflejada en la variedad cromática. La mayoría de los nodos (regulares y élite) provienen de ejecuciones individuales, sin un solapamiento estructural significativo. Al eliminar nodos regulares de inicio, se revelan componentes más conectadas, aunque siguen predominando los nodos propios. Las vistas filtradas evidencian estructuras fragmentadas con escasa cohesión entre regiones élite.

Desde una perspectiva cuantitativa (Tabla 11), el número de arcos es ligeramente inferior al de nodos, pero con magnitudes similares (e.g., 29.213 nodos vs. 25.189 arcos en el escenario de alto presupuesto L_1). Las componentes (co) se mantienen estables entre locaciones de un mismo grupo de presupuesto, mostrando que la granularidad no modifica significativamente la conectividad.

La proporción de nodos compartidos ($p_{n_{sh}}$) varía entre 17.3 % y 25.6 %, siendo más alta en locaciones L_3 . La mayoría de estos nodos son regulares ($p_{n_{shre}} > 0.19$), mientras que los nodos élite compartidos ($p_{n_{shel}}$) apenas superan el 1 %. Los nodos *shared-mixed* ($p_{n_{shmx}}$) son prácticamente marginales, sugiriendo un bajo solapamiento estructural entre ejecuciones.

Los mejores nodos tienen fuerza de entrada reducida ($str_{nbt}^{in} = 0.025$), valor inferior al observado en las redes individuales. Esto es coherente con la dispersión estructural producida por la fusión de múltiples ejecuciones. La mayoría de los escenarios presenta un único camino corto hacia estos nodos ($pt = 1$), aunque en el caso de alto presupuesto con L_3 se identifican 34 trayectorias distintas con longitud promedio de 22.41, lo que sugiere una mejor accesibilidad relativa a ciertas configuraciones destacadas desde distintas partes de la red.

4.2.5. Resultados de PSO-X (PSO-Mix)

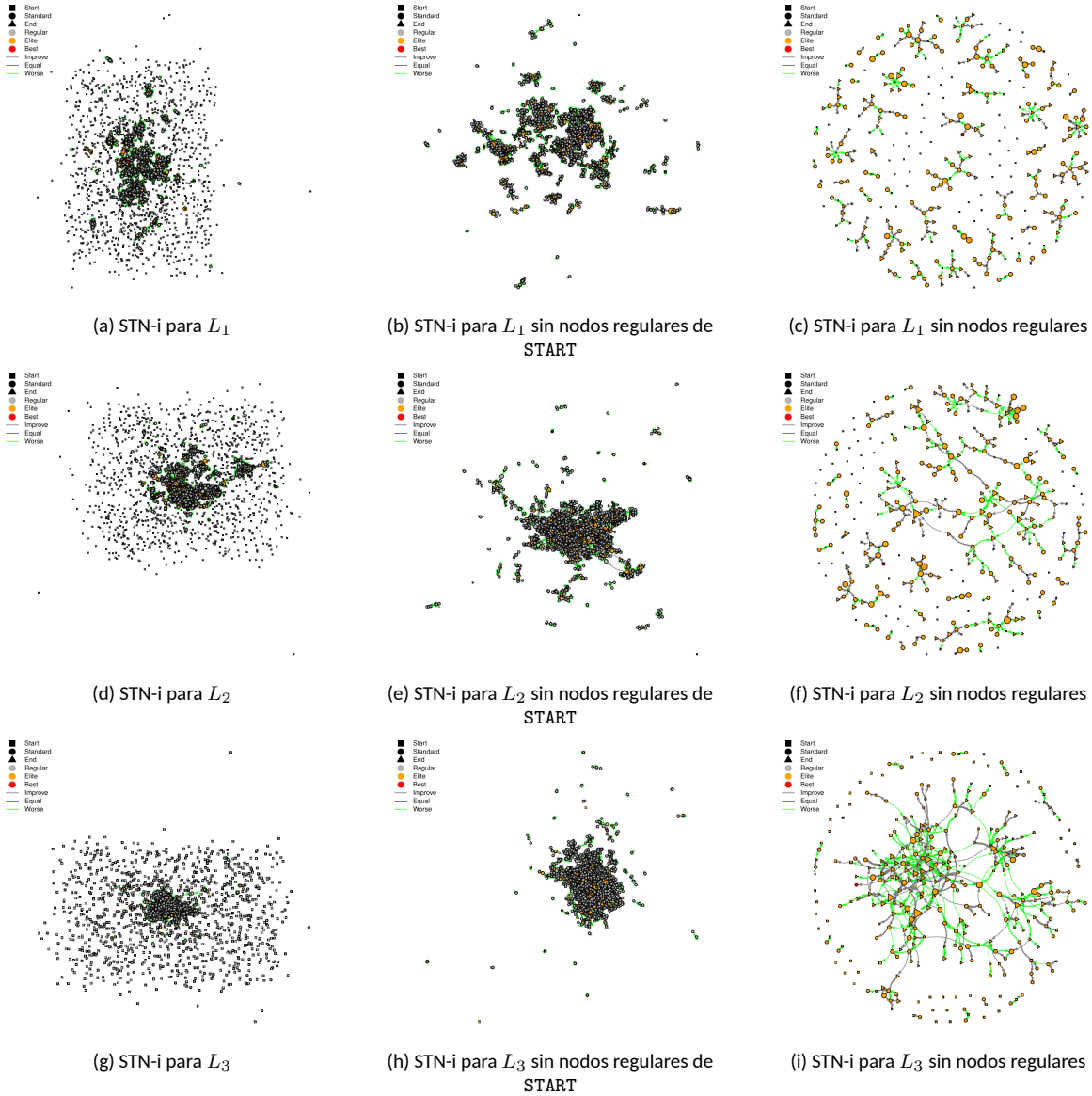


Figura 26: Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia PSO-Mix utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

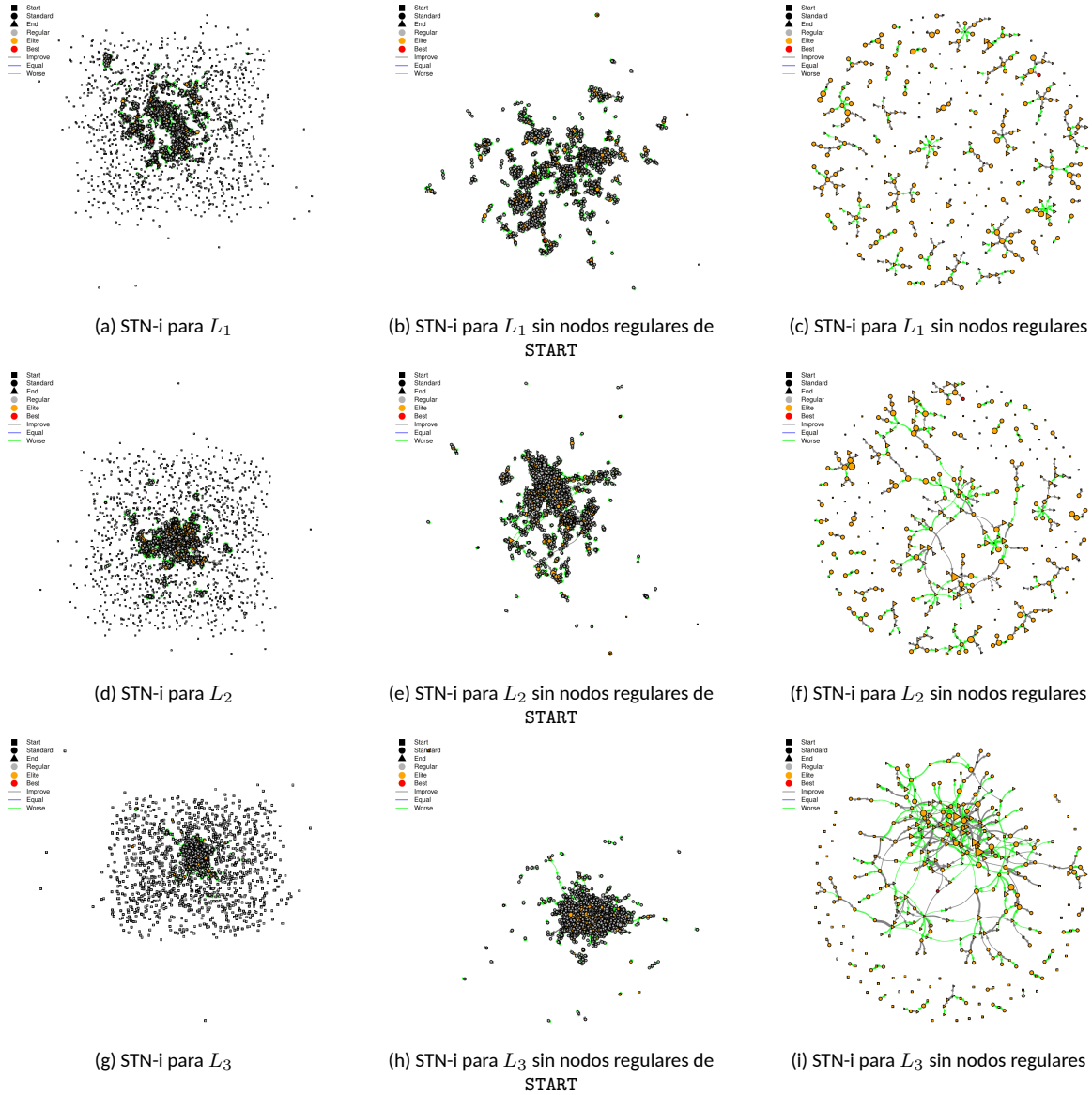


Figura 27: Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia *PSO-Mix* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

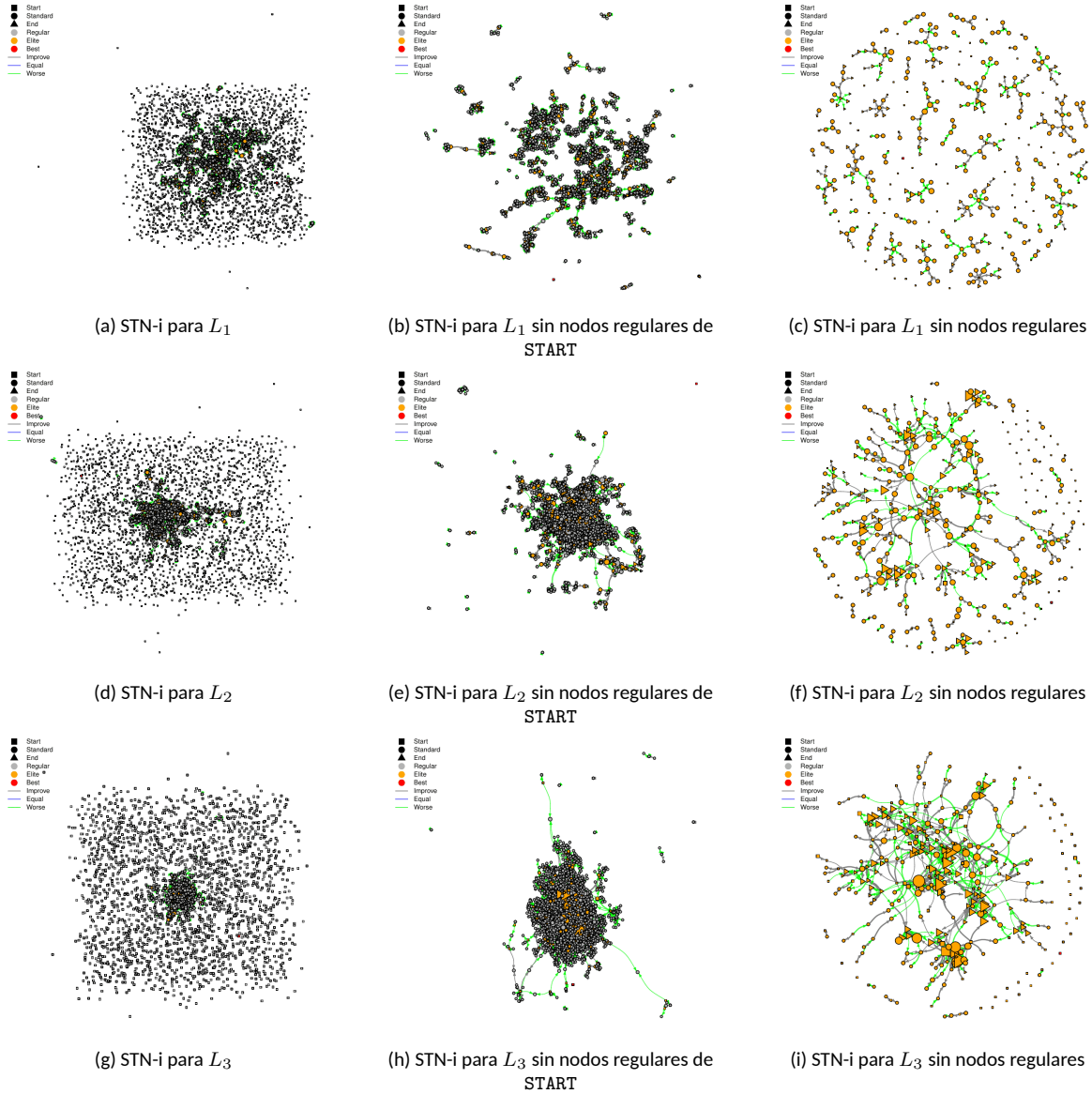


Figura 28: Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia PSO-Mix utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

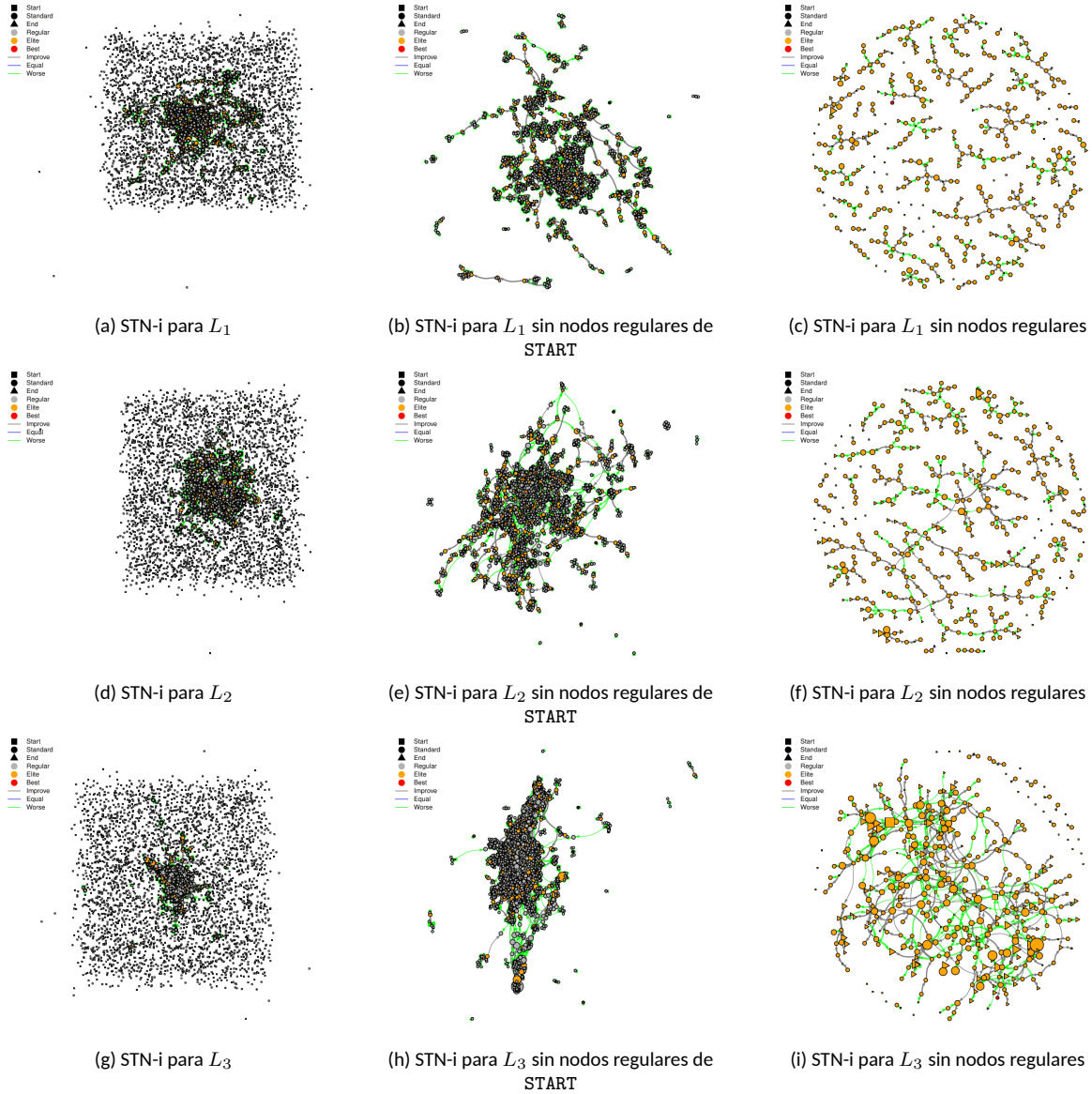


Figura 29: Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia *PSO-Mix* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

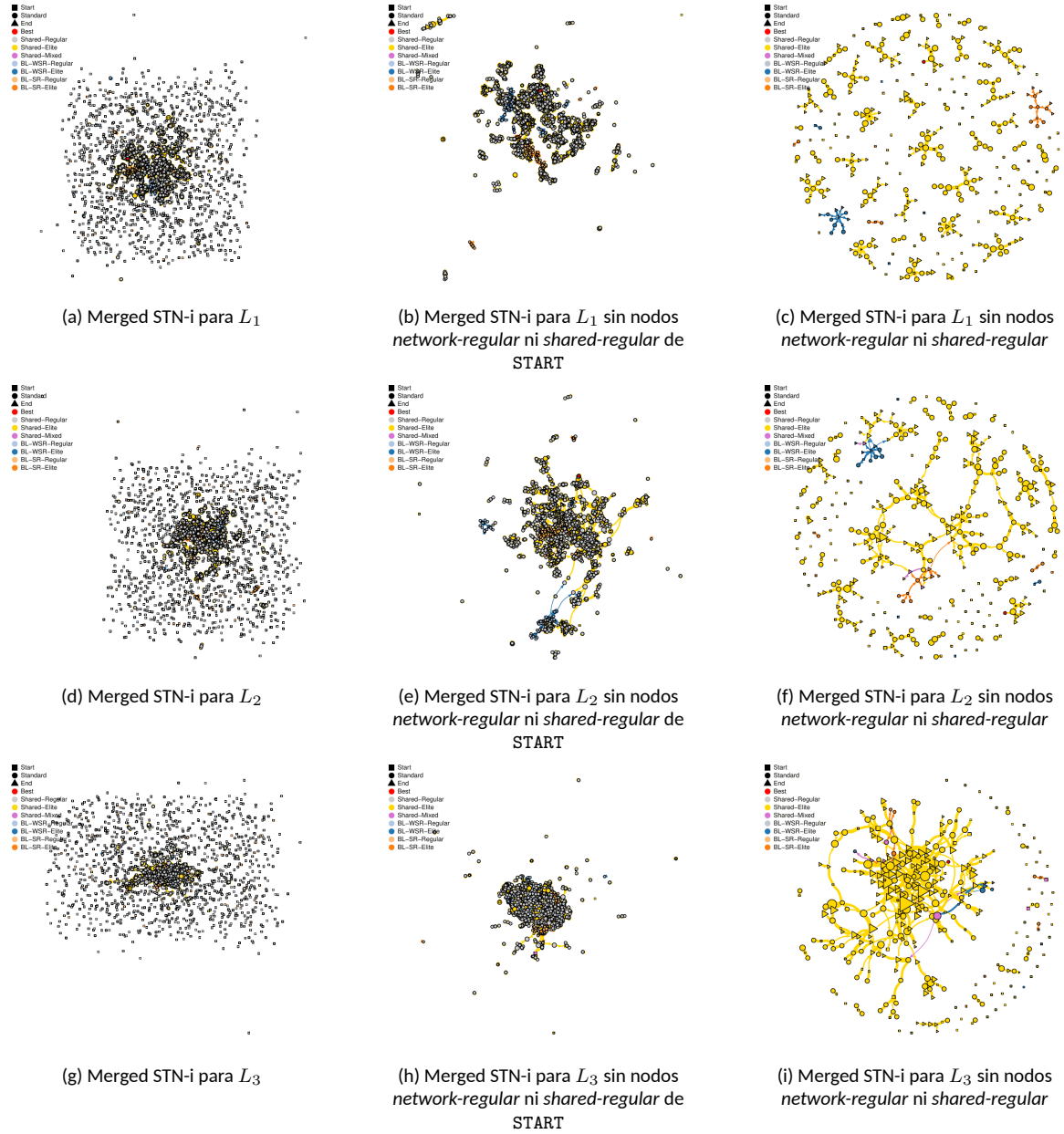


Figura 30: Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia PSO-Mix con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

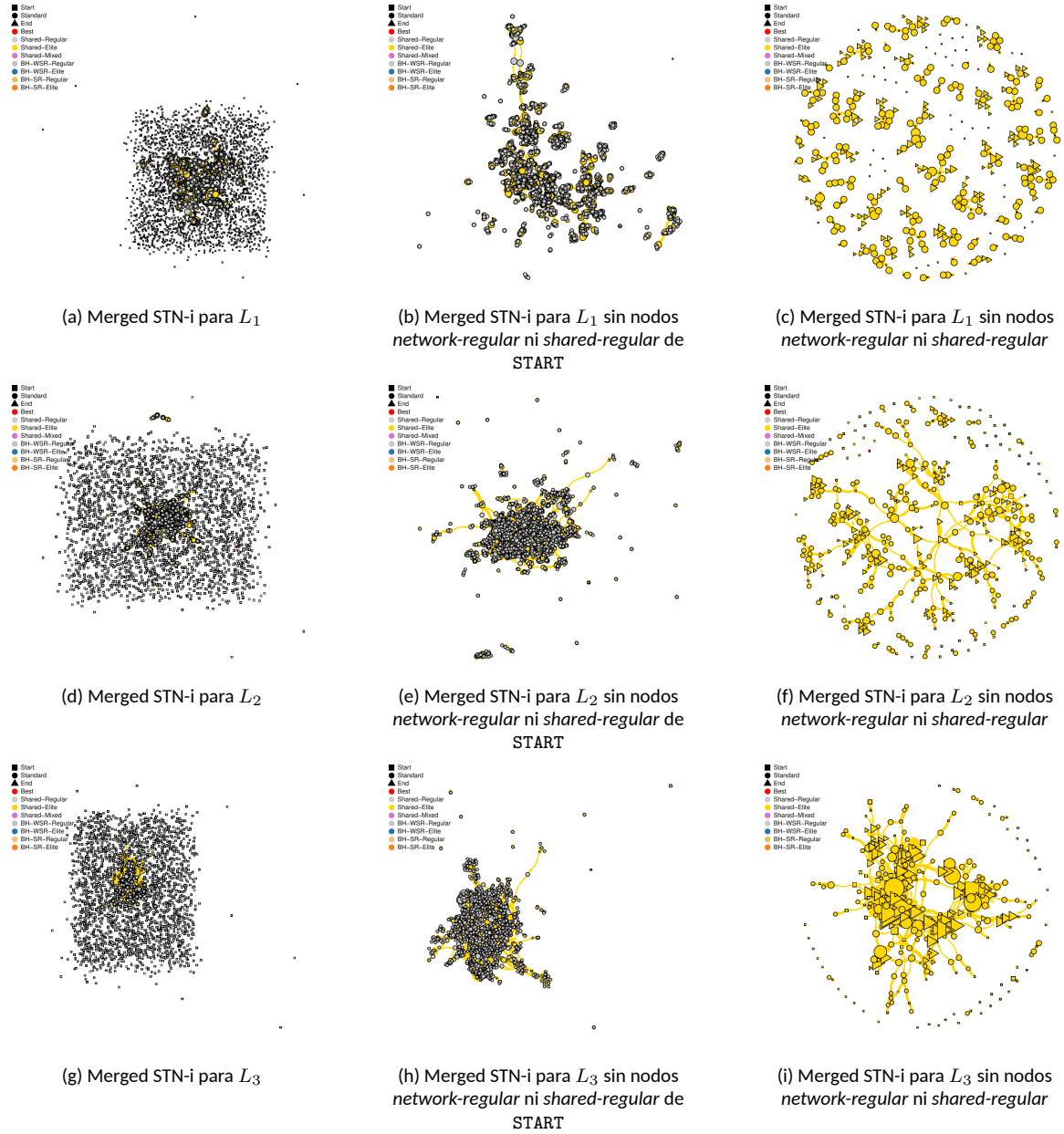


Figura 31: Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia PSO-Mix con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR
IRACE

Escenario	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	ed	$tend_{ed}$	$p_{c-n_{re}}$	$p_{c-n_{el}}$	$p_{c-n_{bt}^{st}}$	$p_{c-n_{el}^{st}}$	$tend_c$	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR	L_1	4657	0.9218	0.0779	0.0002	3339	0.4052	0.8382	0.1618	0.8407	0.1593	0.2118	1319	0.0500	NA	0
	L_2	4472	0.9204	0.0794	0.0002	3279	0.4087	0.8382	0.1618	0.8415	0.1585	0.2118	1279	0.0500	NA	0
	L_3	3843	0.9175	0.0822	0.0003	3058	0.4274	0.8382	0.1618	0.8528	0.1472	0.2118	1192	0.0500	10.5000	4
BL-SR	L_1	4649	0.9211	0.0787	0.0002	3331	0.4017	0.8374	0.1626	0.8390	0.1610	0.2103	1319	0.0500	NA	0
	L_2	4470	0.9199	0.0799	0.0002	3273	0.4073	0.8374	0.1626	0.8396	0.1604	0.2103	1279	0.0500	NA	0
	L_3	3859	0.9173	0.0824	0.0003	3062	0.4233	0.8374	0.1626	0.8491	0.1509	0.2103	1196	0.0500	12.9286	14
BH-WSR	L_1	9384	0.9558	0.0441	0.0001	6736	0.4176	0.9058	0.0942	0.9184	0.0816	0.1701	2653	0.0000	0.0000	1
	L_2	8428	0.9528	0.0471	0.0001	6310	0.4382	0.9058	0.0942	0.9186	0.0814	0.1701	2568	0.0000	0.0000	1
	L_3	6774	0.9510	0.0489	0.0001	5503	0.4641	0.9058	0.0942	0.9259	0.0741	0.1701	2317	0.0000	0.0000	1
BH-SR	L_1	9384	0.9558	0.0441	0.0001	6736	0.4176	0.9058	0.0942	0.9184	0.0816	0.1701	2653	0.0000	0.0000	1
	L_2	8428	0.9528	0.0471	0.0001	6310	0.4382	0.9058	0.0942	0.9186	0.0814	0.1701	2568	0.0000	0.0000	1
	L_3	6774	0.9510	0.0489	0.0001	5503	0.4641	0.9058	0.0942	0.9259	0.0741	0.1701	2317	0.0000	0.0000	1

Tabla 13: Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia *PSO-Mix*.

Escenarios	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	$p_{n_{sh}}$	$p_{n_{shre}}$	$p_{n_{shel}}$	$p_{n_{shmx}}$	ed	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR $\cup$ BL-SR	L_1	4904	0.0948	0.0075	0.0002	0.8976	0.8269	0.0706	0.0000	3520	1385	0.0500	NA	0
	L_2	4699	0.0904	0.0066	0.0002	0.9030	0.8300	0.0721	0.0006	3455	1343	0.0500	NA	0
	L_3	4022	0.0801	0.0050	0.0002	0.9150	0.8376	0.0756	0.0015	3215	1249	0.0500	24.5625	16
BH-WSR $\cup$ BH-SR	L_1	9384	0.0000	0.0000	0.0001	1.0000	0.9558	0.0441	0.0000	6736	2653	0.0000	0.0000	1
	L_2	8428	0.0000	0.0000	0.0001	1.0000	0.9528	0.0471	0.0000	6310	2568	0.0000	0.0000	1
	L_3	6774	0.0000	0.0000	0.0001	1.0000	0.9510	0.0489	0.0000	5503	2317	0.0000	0.0000	1

Tabla 14: Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia *PSO-Mix*.

Escenario	Configuraciones	Iter. Mín	Iter. Máx	Iter. Prom	Uso de <i>soft-restart</i>
BL-WSR	4659	5	7	5.6500	–
BL-SR	4651	5	7	5.6500	0.0000
BH-WSR	9402	5	8	6.2000	–
BH-SR	9402	5	8	6.2000	0.0000

Tabla 15: Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia *PSO-Mix* del algoritmo PSO-X.

4.2.6. Análisis de PSO-X (PSO-Mix)

El conjunto de funciones *PSO-Mix* está compuesto por funciones híbridas y de composición extraídas de distintas competencias estándar (CEC'05, CEC'14, SOCO'10), todas definidas en un espacio de 30 dimensiones. Estas funciones presentan una alta complejidad estructural, producto de la combinación de múltiples subcomponentes con distintos grados de dificultad y regularidad, generando paisajes de optimización altamente irregulares, no lineales y propensos a bifurcaciones abruptas. Dicha complejidad impacta directamente la dinámica de exploración de los algoritmos y, en consecuencia, la variabilidad de las trayectorias generadas por *irace*.

En este contexto, PSO-X se configura como un *framework* particularmente desafiante de sintonizar. Este cuenta con un total de 8 parámetros, incluyendo 2 condicionales que se activan bajo ciertas topologías. De estos, 6 fueron discretizados para construir las STN-i, mientras que los 2 restantes, de tipo categórico, se mantuvieron en su forma original. Esta estructura genera un espacio paramétrico altamente combinatorio y denso, en el que resulta difícil obtener trayectorias repetidas o agrupamientos consistentes de configuraciones, especialmente ante funciones con elevada sensibilidad a la parametrización, como ocurre en este conjunto.

En consecuencia, se anticipa que las STN-i generadas a partir de este dominio presenten una estructura dispersa, con alta fragmentación, baja conectividad y una fuerte presencia de nodos individuales que representan configuraciones descartadas tempranamente por *irace*. Estas condiciones también limitan parcialmente el potencial de compresión estructural de las discretizaciones más agresivas (L_2 y L_3), lo que convierte a este caso en un referente relevante para evaluar la robustez y sensibilidad de las STN-i ante espacios de búsqueda complejos.

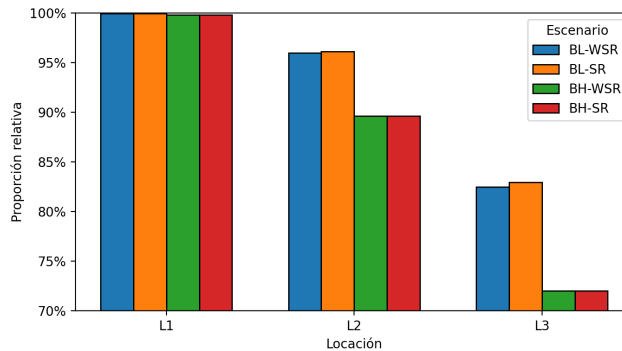


Figura 32: Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia *PSO-Mix* del algoritmo PSO-X.

La Figura 32 evidencia que las locaciones L_1 y L_2 mantienen proporciones superiores al 90 % en todos los escenarios, lo que indica una escasa capacidad de agrupamiento entre configuraciones. Esta tendencia se mantiene en escenarios de mayor presupuesto, donde el incre-

mento en el número de configuraciones evaluadas no conlleva una mejora estructural sustancial en términos de reducción de nodos, sino que exhibe solo una leve disminución en la proporción relativa. Esto sugiere que el efecto de la granularidad en el agrupamiento se mantiene limitado, aunque con una tendencia ligeramente más favorable en escenarios con mayor presupuesto.

En contraste, la locación L_3 muestra una reducción más marcada, alcanzando proporciones cercanas al 80 % en bajo presupuesto y a 72 % en alto, lo cual sugiere un mayor poder de síntesis. Esta compresión evidencia que una granularidad más gruesa comienza a capturar regiones estructuralmente consistentes dentro del espacio, donde configuraciones distintas caen en locaciones comunes, lo que facilita la identificación de patrones repetidos por parte de las STN-i.

La Tabla 15 indica que el número de iteraciones permanece estable en todos los escenarios, con un promedio entre 5.65 y 6.20. Esto revela que el incremento del presupuesto se traduce en una mayor cantidad de configuraciones evaluadas por iteración, manteniendo constante la dinámica global del proceso. Este fenómeno tiene un efecto relevante: al aumentar el volumen de evaluación por iteración, también se incrementa la presión selectiva, lo que tiende a reducir la diversidad y limitar la permanencia de configuraciones de calidad.

A pesar de ello, el mecanismo *soft-restart* no se activa cuando está habilitado. Esto podría explicarse por la ausencia de patrones claros de estancamiento, o por la complejidad del dominio, que mantiene la diversidad sin necesidad de reinicio artificial.

A nivel estructural, las STN-i de *PSO-Mix* (Figuras 26 a 29, Tabla 13) muestran una predominancia abrumadora de nodos regulares ($p_{n_{re}} > 91\%$), que se eleva a más del 95 % en escenarios de alto presupuesto. Esto indica que la mayor parte del espacio de búsqueda es descartado tempranamente respecto a su generación. Esta observación visual se refleja en las subfiguras (a), (d) y (g), donde predominan los nodos grises desconexos que representan configuraciones eliminadas. La baja proporción de nodos élite ($p_{n_{el}}$ entre 5 % y 8 %) y de nodos con mejores configuraciones ($p_{n_{bt}} \approx 0.01\%$) refuerza la hipótesis de que *irace* identifica regiones prometedoras en etapas avanzadas, sin converger de forma agresiva hacia un conjunto reducido de soluciones.

La composición interna de los nodos muestra un patrón consistente: las configuraciones regulares representan más del 83 % del total, incluso en nodos de inicio. Las configuraciones élite son escasas y, cuando aparecen, lo hacen principalmente en etapas posteriores, como se evidencia en las subfiguras (c), (f) y (i) de las mismas figuras, donde los nodos élite aparecen aislados y desconectados. Esta evidencia es coherente con un proceso de sintonización dominado por el descarte temprano y la exploración amplia, donde solo una pequeña fracción del espacio se retiene para explotación.

Según la Tabla 13, en los escenarios de bajo presupuesto, la fuerza de entrada hacia los mejores nodos es marginal ($str_{n_{bt}}^{in} = 0.05$) y no se detectan trayectorias entrantes en L_1 y L_2 ($pt = 0$), lo que indica que los mejores nodos corresponden a configuraciones de inicio sin

conexión estructural aparente. Solo en L_3 se identifican trayectorias hacia estos nodos (4 caminos, longitud promedio de 10.5), lo que demuestra que una mayor granularidad favorece la aparición de rutas conectadas hacia regiones de buen rendimiento.

En alto presupuesto, en cambio, los mejores nodos no presentan trayectorias de acceso ($str_{nbt}^{in} = 0$, $pt = 1$), sugiriendo que las configuraciones de alto desempeño emergen de forma aislada o son rápidamente descartadas, impidiendo que se estructuren trayectorias hacia ellas.

La métrica de supervivencia de configuraciones élite reportada en la Tabla 13 ($tend_c$) también refleja el efecto del presupuesto. En escenarios de bajo presupuesto se observa un valor de 0.21, mientras que en alto este desciende a 0.17. Esto sugiere que con menos configuraciones por iteración, las soluciones de calidad tienen más oportunidades de mantenerse en el proceso, dado que la presión selectiva es menor y permite una explotación más prolongada de regiones promisorios.

Las merged STN-i conservan una alta proporción de nodos de inicio descartados, reflejo de la exploración inicial agresiva de irace, como se aprecia en las vistas completas (subfiguras (a), (d), (g)) de la Figura 30 y la Figura 31. Al filtrar estos nodos, emergen estructuras más compactas en las subfiguras (b), (e), (h), donde las conexiones entre nodos relevantes se vuelven más evidentes. Este efecto es más marcado en L_3 (subfigura (h)), donde los nodos centrales comienzan a actuar como puntos de convergencia, evidenciando una consolidación progresiva de trayectorias y agrupamientos más densos y visualmente organizados. En contraste, las locaciones L_1 y L_2 (subfiguras (b) y (e)) conservan una fragmentación elevada, con múltiples subcomponentes desconexos que reflejan una mayor dispersión del espacio explorado.

La Tabla 14 muestra que la proporción de nodos compartidos es notablemente alta en todos los escenarios ($p_{n_{sh}}$ entre 89.7% y 100%). Esta coincidencia es especialmente interesante considerando la complejidad y aleatoriedad del espacio de búsqueda. El hecho de que distintas ejecuciones converjan hacia nodos comunes —mayoritariamente regulares ($p_{n_{shre}} > 75\%$)— sugiere que irace explora consistentemente ciertas regiones descartadas, independientemente de la semilla utilizada.

Los nodos élite compartidos ($p_{n_{shel}}$) se mantienen bajos (4–7%), indicando que la calidad estructural no siempre es consistente entre ejecuciones. Solo en escenarios de bajo presupuesto aparecen nodos mixtos compartidos ($p_{n_{shmx}} > 0$), lo cual indica cierta variabilidad en la clasificación estructural según el contexto local de la trayectoria.

De acuerdo con la Tabla 14, la accesibilidad hacia los mejores nodos es limitada. Solo en el escenario de bajo presupuesto con locación L_3 se identifican 16 trayectorias de acceso, con longitud promedio de 24.56, lo que sugiere una síntesis estructural valiosa: estos caminos se construyen a partir de fragmentos dispersos en las ejecuciones originales, generando un recorrido más profundo hacia regiones de alto rendimiento. En el resto de escenarios, los mejores nodos permanecen aislados, sin caminos efectivos desde el resto de la red.

4.2.7. Resultados de PSO-X (PSO-Mul)

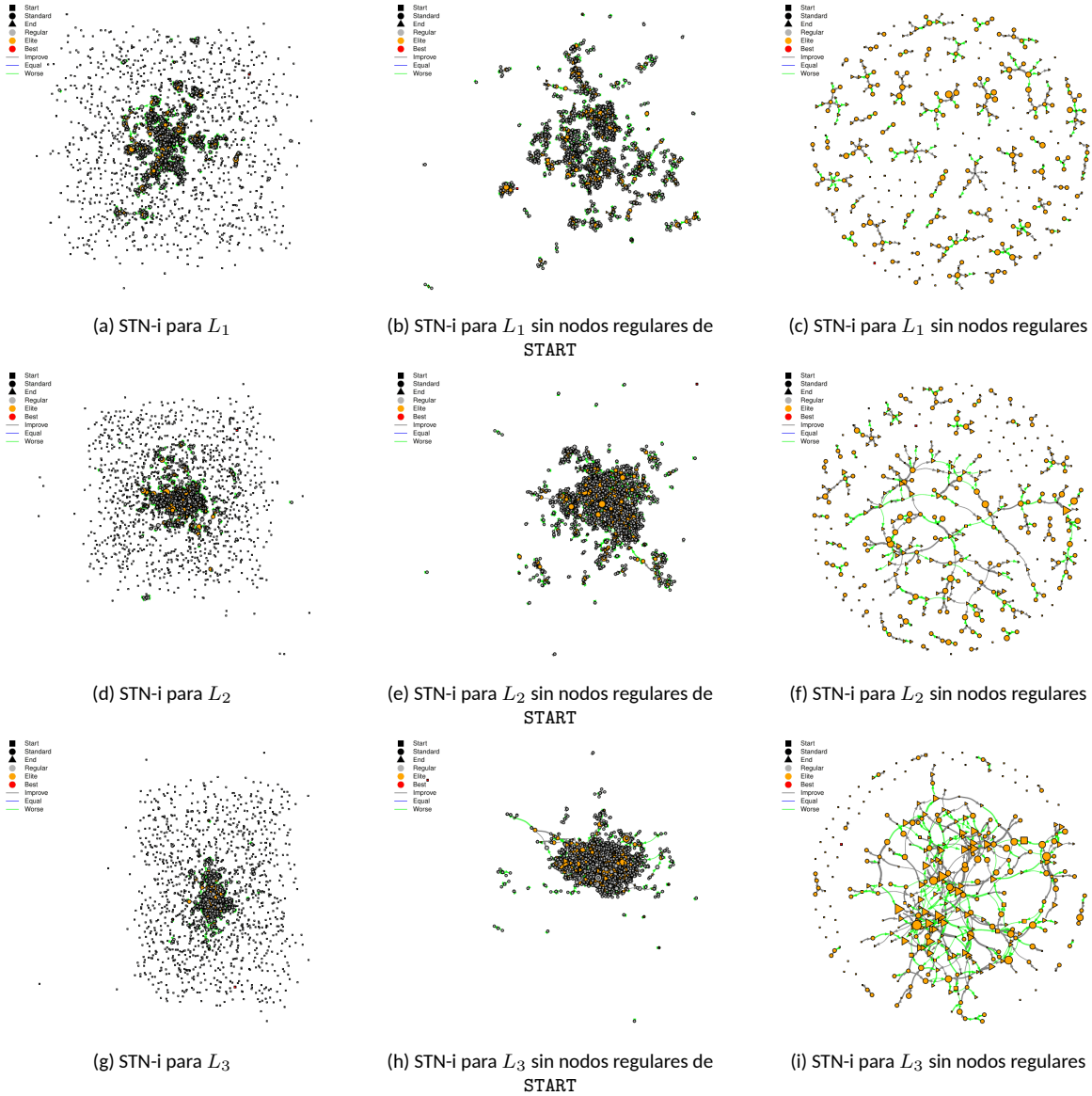


Figura 33: Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia *PSO-Mul* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

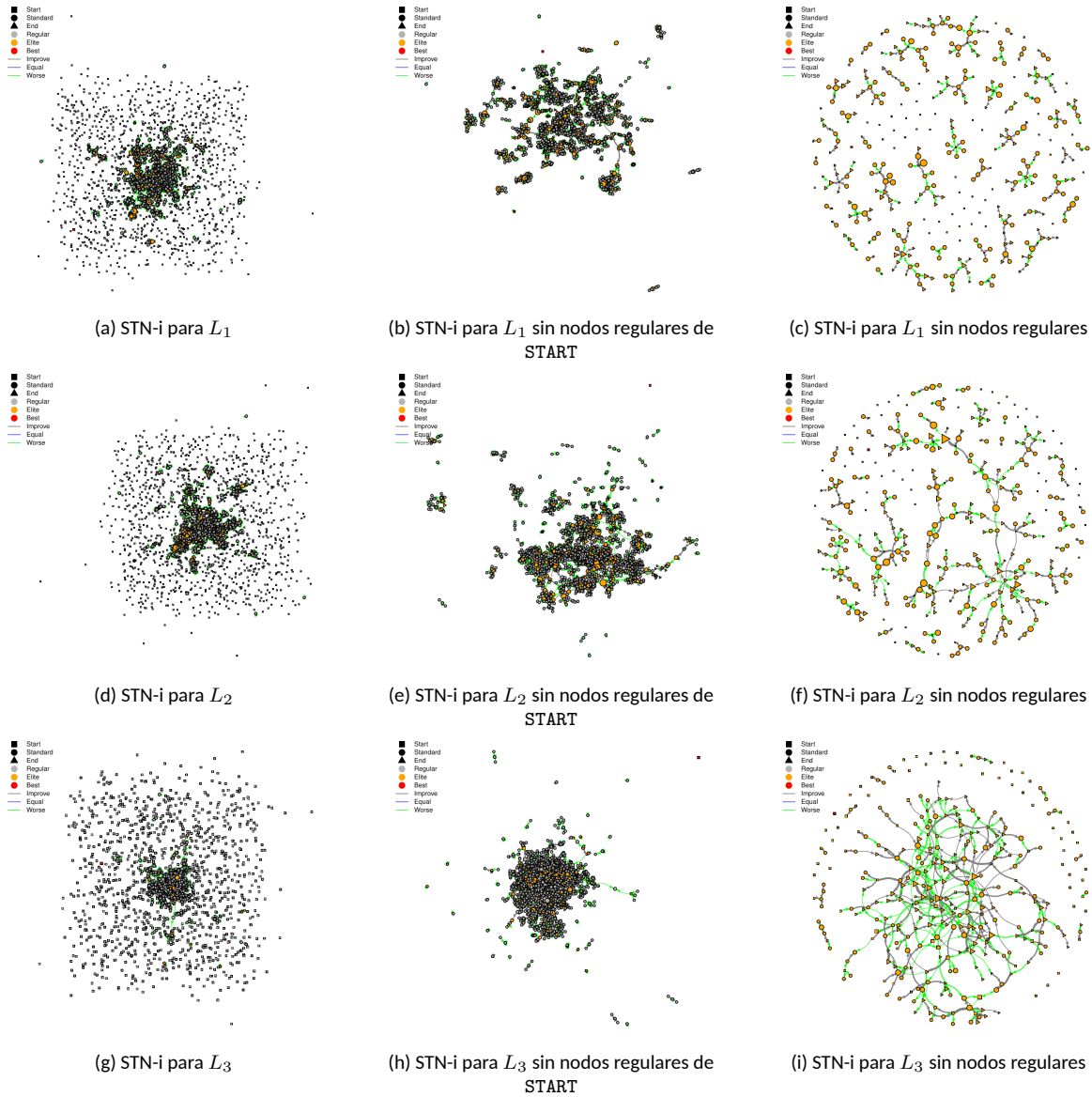


Figura 34: Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia *PSO-Mul* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

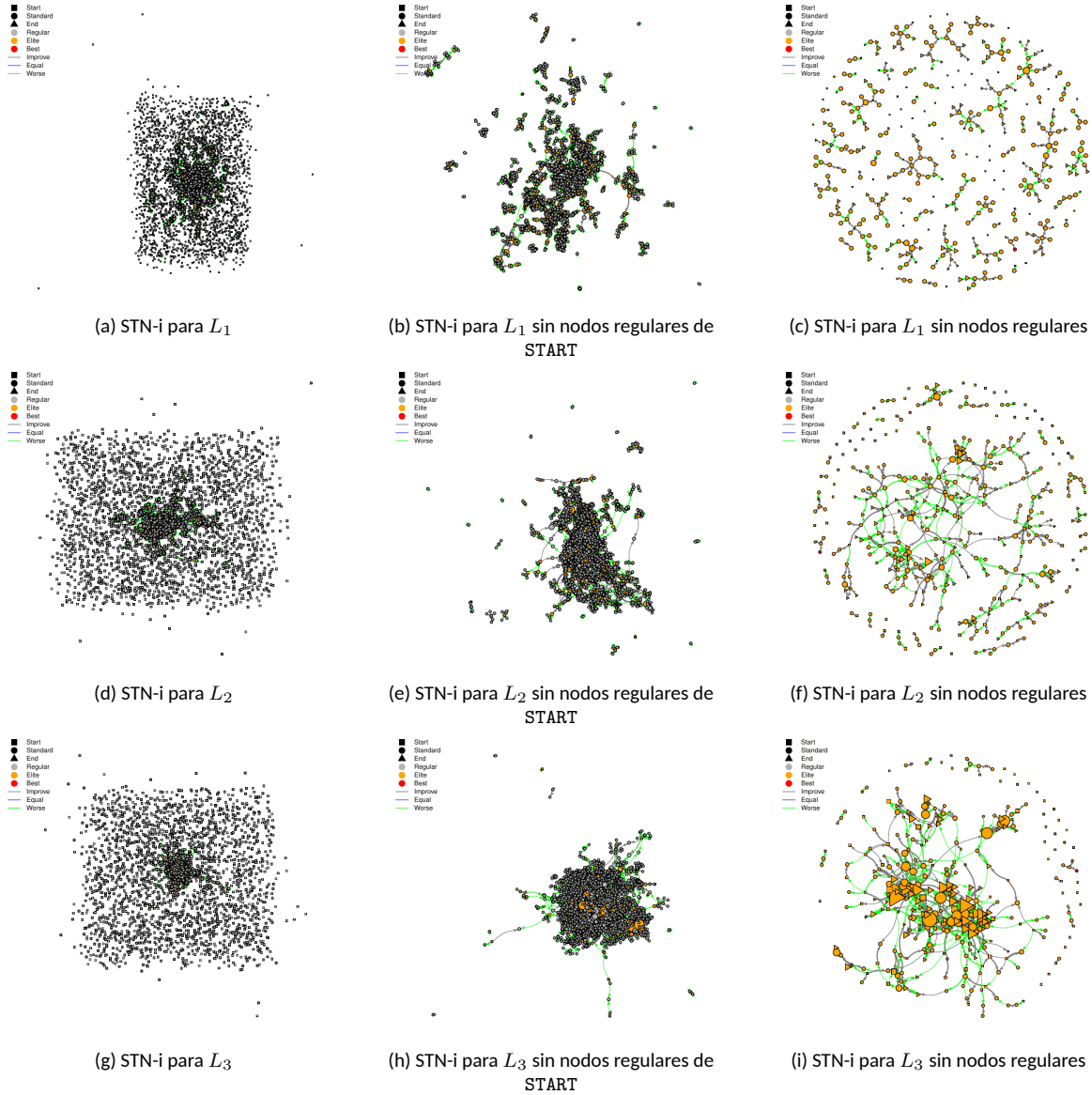


Figura 35: Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia *PSO-Mul* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

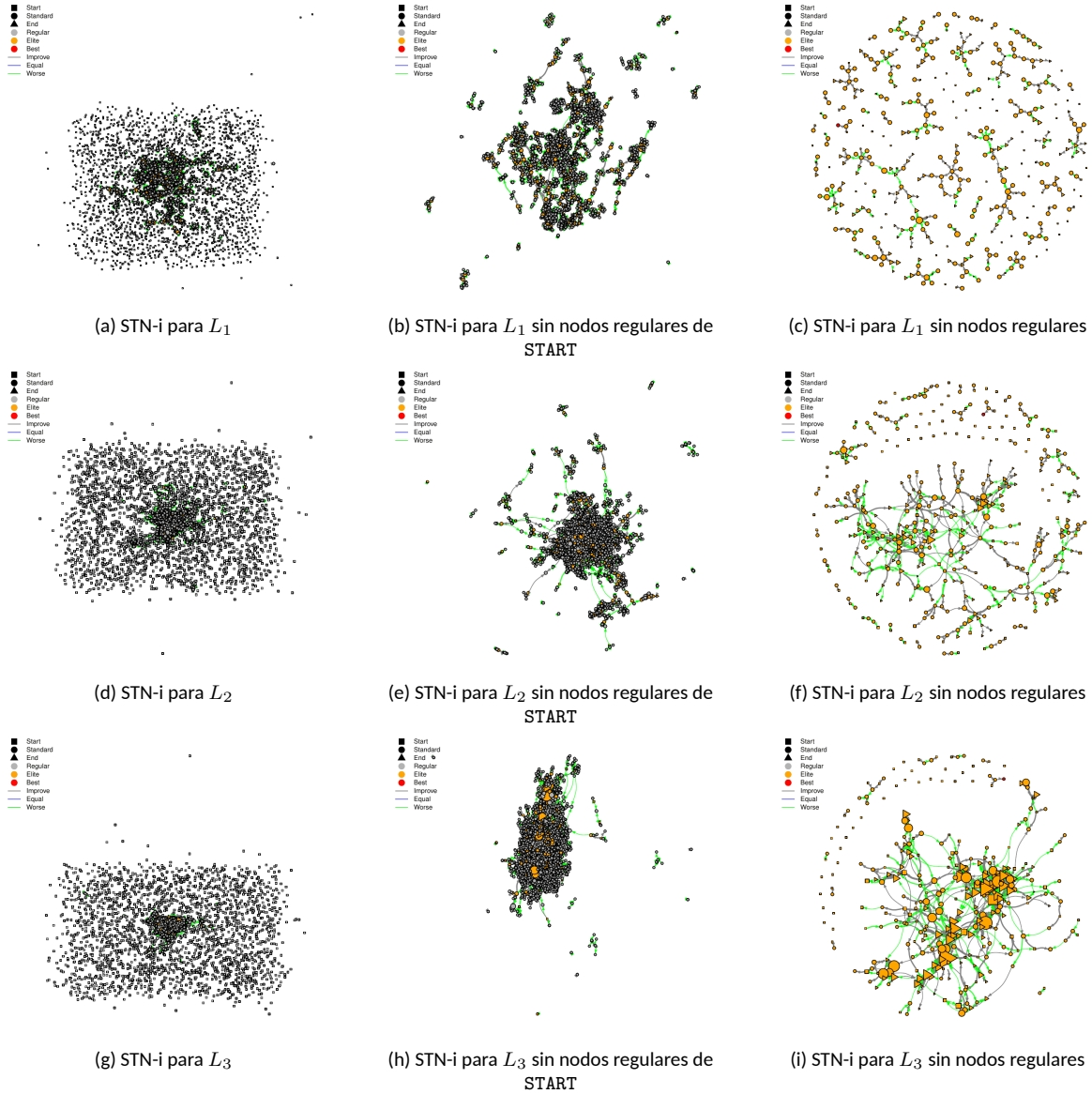


Figura 36: Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia *PSO-Mul* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

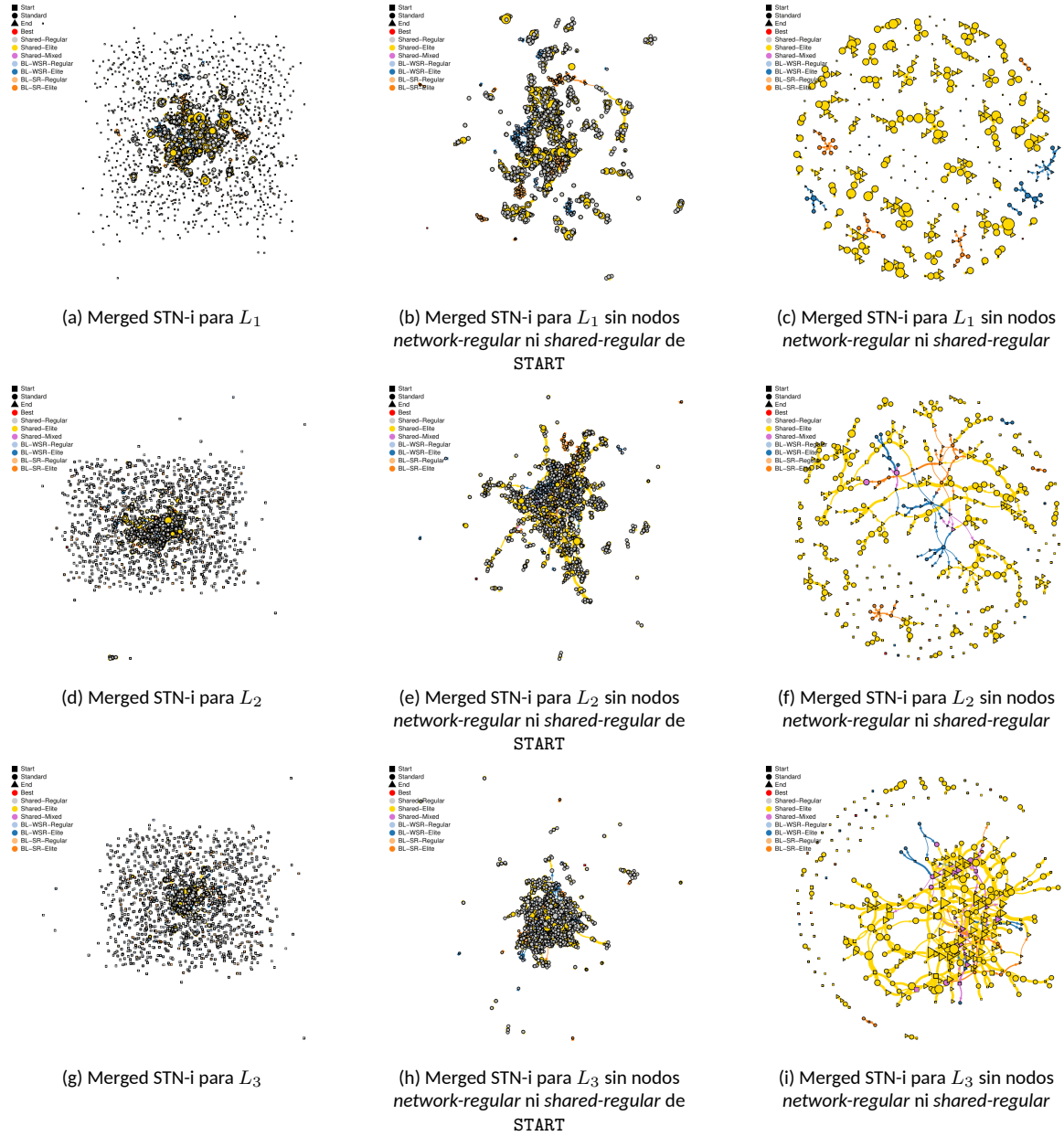


Figura 37: Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia *PSO-Mul* con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

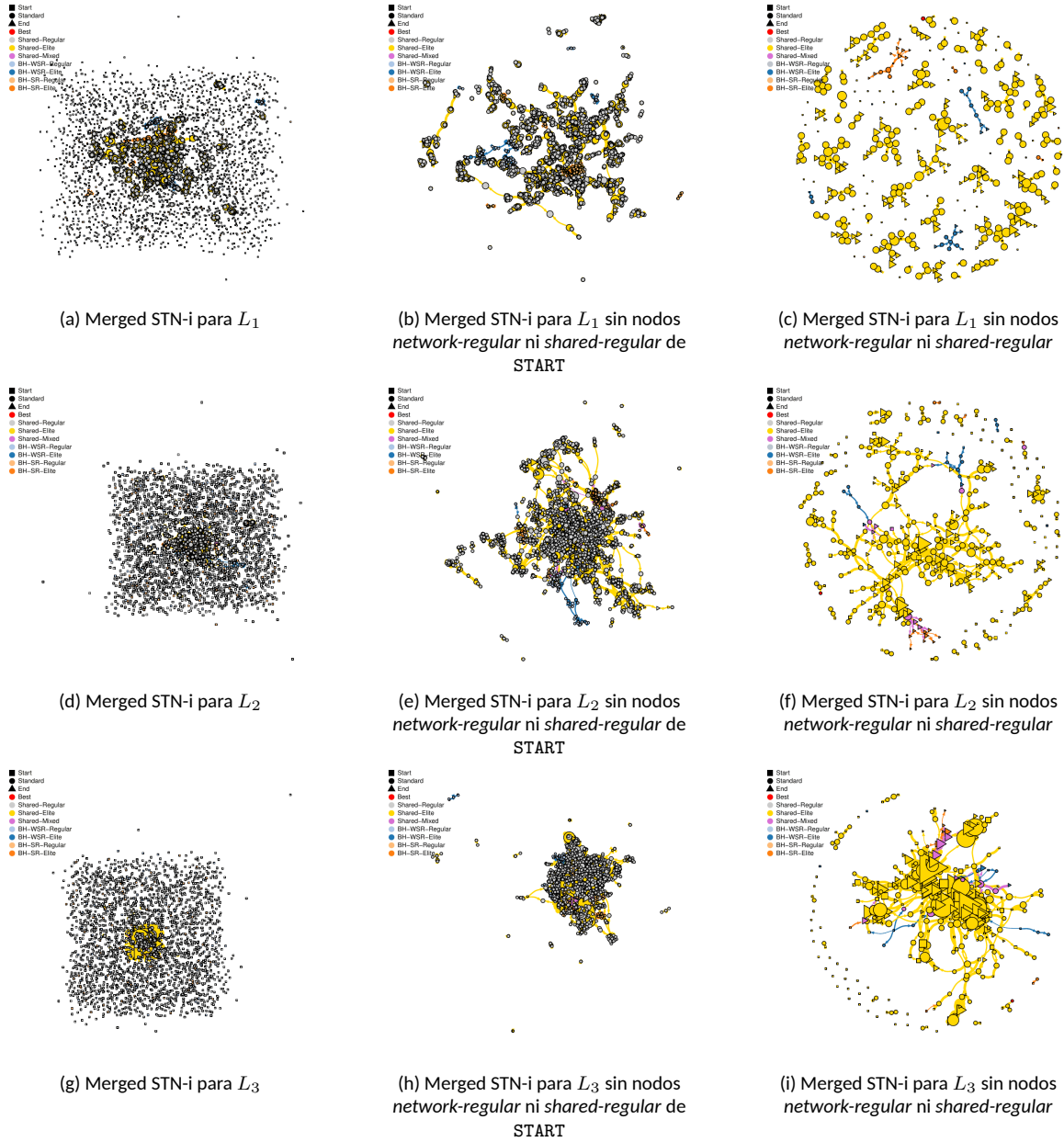


Figura 38: Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia *PSO-Mul* con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR
IRACE

Escenario	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	ed	$tend_{ed}$	$p_{c-n_{re}}$	$p_{c-n_{el}}$	$p_{c-n_{re}^t}$	$p_{c-n_{el}^t}$	$tend_c$	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR	L_1	4592	0.9183	0.0814	0.0002	3274	0.4020	0.8311	0.1689	0.8312	0.1688	0.2355	1318	0.0000	0.0000	1
	L_2	4431	0.9167	0.0831	0.0002	3221	0.4011	0.8311	0.1689	0.8339	0.1661	0.2355	1276	0.0000	0.0000	1
	L_3	3897	0.9151	0.0847	0.0003	3038	0.4006	0.8311	0.1689	0.8414	0.1586	0.2355	1208	0.0000	0.0000	1
BL-SR	L_1	4593	0.9190	0.0808	0.0002	3276	0.4057	0.8309	0.1691	0.8347	0.1653	0.2392	1317	0.0000	0.0000	1
	L_2	4440	0.9178	0.0820	0.0002	3226	0.4067	0.8309	0.1691	0.8373	0.1627	0.2392	1278	0.0000	0.0000	1
	L_3	3900	0.9164	0.0833	0.0003	3040	0.4128	0.8309	0.1691	0.8443	0.1557	0.2392	1217	0.0000	0.0000	1
BH-WSR	L_1	9649	0.9561	0.0438	0.0001	7014	0.4266	0.9044	0.0956	0.9140	0.0860	0.1786	2643	0.0500	3.0000	1
	L_2	8489	0.9537	0.0462	0.0001	6479	0.4203	0.9044	0.0956	0.9145	0.0855	0.1786	2560	0.0500	3.0000	1
	L_3	6838	0.9528	0.0471	0.0001	5571	0.3965	0.9044	0.0956	0.9192	0.0808	0.1786	2369	0.0500	3.0000	1
BH-SR	L_1	9622	0.9562	0.0436	0.0001	6988	0.4283	0.9049	0.0951	0.9134	0.0866	0.1749	2643	0.0500	3.0000	1
	L_2	8449	0.9540	0.0459	0.0001	6456	0.4287	0.9049	0.0951	0.9153	0.0847	0.1749	2560	0.0500	3.0000	1
	L_3	6785	0.9530	0.0469	0.0001	5533	0.4038	0.9049	0.0951	0.9204	0.0796	0.1749	2367	0.0500	3.0000	1

Tabla 16: Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia *PSO-Mul*.

Escenarios	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	$p_{n_{sh}}$	$p_{n_{shre}}$	$p_{n_{shel}}$	$p_{n_{shmx}}$	ed	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR $\cup$ BL-SR	L_1	5041	0.1635	0.0145	0.0002	0.8221	0.7552	0.0667	0.0000	3593	1448	0.0000	0.0000	1
	L_2	4848	0.1568	0.0134	0.0002	0.8298	0.7603	0.0683	0.0010	3537	1403	0.0000	0.0000	1
	L_3	4232	0.1493	0.0083	0.0002	0.8424	0.7668	0.0711	0.0043	3334	1329	0.0000	0.0000	1
BH-WSR $\cup$ BH-SR	L_1	10100	0.0877	0.0043	0.0001	0.9080	0.8683	0.0396	0.0000	7334	2775	0.0500	6.0000	1
	L_2	8874	0.0877	0.0036	0.0001	0.9087	0.8659	0.0416	0.0011	6784	2685	0.0500	6.0000	1
	L_3	7116	0.0822	0.0034	0.0001	0.9144	0.8703	0.0426	0.0014	5824	2478	0.0500	6.0000	1

Tabla 17: Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia *PSO-Mul*.

Escenario	Configuraciones	Iter. Mín	Iter. Máx	Iter. Prom	Uso de <i>soft-restart</i>
BL-WSR	4595	5	7	5.8500	–
BL-SR	4597	5	7	5.9000	0.0000
BH-WSR	9679	5	8	6.5500	–
BH-SR	9653	5	8	6.5000	0.0000

Tabla 18: Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia *PSO-Mul* del algoritmo PSO-X.

4.2.8. Análisis de PSO-X (PSO-Mul)

El conjunto *PSO-Mul* está compuesto por funciones multimodales de CEC'05, CEC'14 y SOCO'10, todas en 30 dimensiones. Estas funciones se caracterizan por presentar múltiples óptimos locales, lo que demanda del algoritmo un comportamiento altamente exploratorio. Esta característica se potencia al considerar la complejidad inherente al espacio paramétrico de PSO-X, con interacciones condicionales entre parámetros y una codificación parcialmente categórica. El resultado es un espacio de búsqueda altamente heterogéneo y difícil de recorrer de forma sistemática, donde los patrones estructurales son menos evidentes y las configuraciones de calidad pueden estar dispersas.

A nivel de discretización, se emplea la misma estrategia que en *PSO-Mix*, discretizando 6 parámetros sobre los 8 disponibles, lo que da lugar a un espacio de locaciones complejo pero comparable en términos de granularidad. Este conjunto se torna especialmente relevante para evaluar cómo la fragmentación y la conectividad de las STN-i se comportan cuando el dominio requiere exploración diversificada y adaptativa.

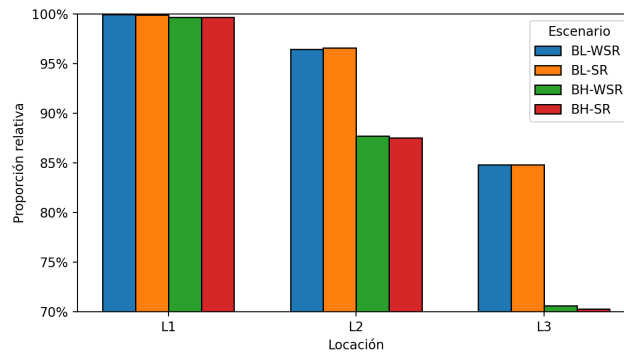


Figura 39: Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia *PSO-Mul* del algoritmo PSO-X.

La Figura 39 revela una tendencia similar a la observada en *PSO-Mix*, con proporciones cercanas al 100 % en L_1 y superiores al 90 % en L_2 , aunque con una caída más pronunciada en L_3 , particularmente en escenarios de alto presupuesto, donde se alcanzan valores cercanos al 70 %. Esta reducción indica que, a mayor granularidad, las locaciones logran sintetizar de forma más efectiva regiones comunes, aunque el efecto sigue siendo limitado en comparación con dominios de menor complejidad como QAP-60. Esto refuerza la noción de que, en espacios multimodales, la dispersión estructural dificulta la consolidación de regiones exploradas de manera consistente.

La Tabla 18 muestra que el número de iteraciones se mantiene relativamente estable (entre 5.85 y 6.55), lo que sugiere que el incremento del presupuesto impacta principalmente en el número de configuraciones por iteración. El mecanismo *soft-restart* no se activa en ninguno de los escenarios, lo cual es coherente con la alta diversidad del espacio y con la ausencia de señales de convergencia prematura. La capacidad del algoritmo de mantener trayectorias

suficientemente variadas podría evitar la necesidad de reinicio, aunque esto no implica una mejora estructural significativa en la red generada.

Las visualizaciones de las STN-i (Figuras 33 a 36) revelan una alta concentración de nodos regulares, especialmente en locaciones L_1 y L_2 , donde la fragmentación estructural es evidente. La eliminación de nodos de inicio permite revelar subredes más conectadas, particularmente en L_3 , que comienza a mostrar una estructura más integrada del espacio explorado.

Las métricas de conectividad evidencian este comportamiento: el número de componentes conectadas disminuye levemente con la granularidad, mientras que el número de arcos por nodo mejora solo marginalmente. En general, el valor de $tend_{ed}$ se mantiene bajo, indicando una direccionalidad débil en las transiciones entre configuraciones.

Como se detalla en la Tabla 16, las STN-i de *PSO-Mul* presentan proporciones de nodos regulares muy altas ($p_{n_{re}} > 91\%$ en bajo presupuesto y hasta 96% en alto), lo que indica una política agresiva de descarte por parte de *irace*. La proporción de nodos élite es baja ($4-8\%$), y los nodos con mejores configuraciones ($p_{n_{bt}}$) son prácticamente marginales. Esta tendencia también se observa en la composición de los nodos, donde las configuraciones regulares superan el 83% incluso en nodos de inicio. En todos los casos, las configuraciones élite tienen baja representación al comienzo del proceso, lo que indica que son descubiertas gradualmente, conforme avanza la sintonización.

De acuerdo con las métricas presentadas en la Tabla 16, en escenarios de bajo presupuesto, los mejores nodos tienden a ser nodos de inicio sin trayectorias asociadas ($str_{n_{bt}}^{in} = 0$, $pt = 1$), lo que sugiere que estas configuraciones de alto rendimiento son seleccionadas tempranamente pero descartadas antes de generar rutas estructuradas hacia ellas. En alto presupuesto, en cambio, se observan trayectorias conectadas con longitud promedio de 3, lo que refleja una mejor conectividad y un comportamiento más articulado del proceso de exploración. Aun así, estas trayectorias siguen siendo limitadas en cantidad y profundidad.

Tal como se resume en la Tabla 16, la métrica de supervivencia $tend_c$ muestra valores cercanos a 0.23 en bajo presupuesto y desciende a 0.17 en alto, reproduciendo el patrón ya observado en *PSO-Mix*. Este comportamiento refuerza la idea de que, al aumentar el volumen de configuraciones por iteración, la presión selectiva se intensifica, reduciendo la permanencia de soluciones élite entre iteraciones y acelerando el descarte.

La Tabla 17 confirma que la proporción de nodos compartidos ($p_{n_{sh}}$) es alta en todos los escenarios (superior al 82%), aunque levemente inferior a la observada en *PSO-Mix*. La mayoría de estos nodos son regulares ($p_{n_{shre}} > 75\%$), mientras que los nodos élite compartidos se mantienen bajos (alrededor del 5%). A medida que aumenta la granularidad, se observa un leve incremento de nodos mixtos ($p_{n_{shmx}}$), lo que puede interpretarse como una mayor ambigüedad en la clasificación estructural de ciertas regiones.

Según los datos presentados en la Tabla 17, la estructura de caminos hacia los mejores nodos mejora en los escenarios fusionados de alto presupuesto, donde se duplica la longitud

promedio de trayectorias (6), con un mayor número de rutas disponibles. Este efecto es consistente con el proceso de fusión, que permite conectar fragmentos de trayectorias originalmente disjuntos, consolidando accesos hacia configuraciones de alta calidad. Sin embargo, estos nodos siguen siendo escasos, y su conectividad, aunque mejorada, aún no es dominante en la estructura global de la red.

4.2.9. Resultados de PSO-X (PSO-Uni)

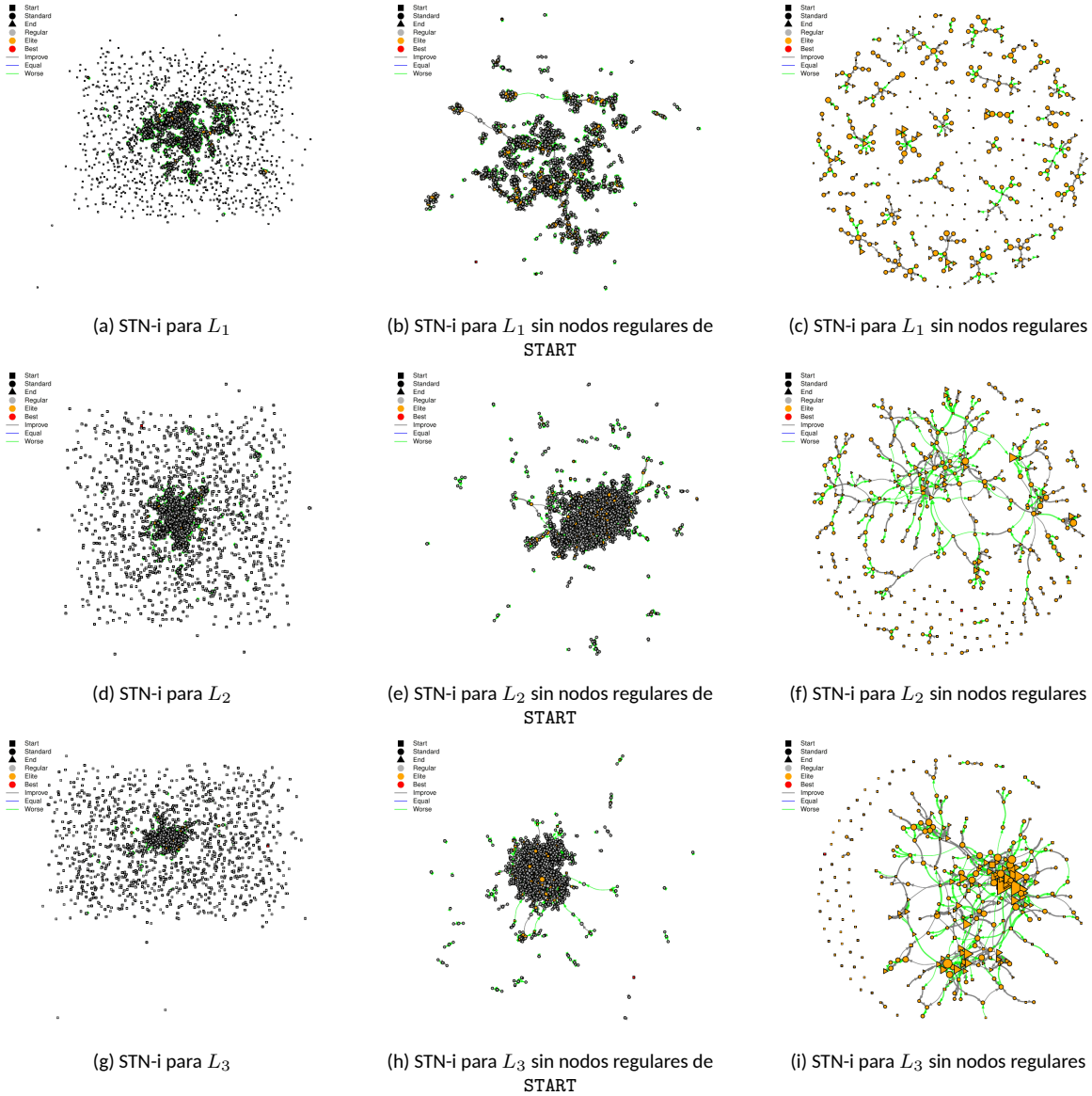


Figura 40: Visualización de las STNs-i generadas a partir del escenario BL-WSR para la instancia *PSO-Uni* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

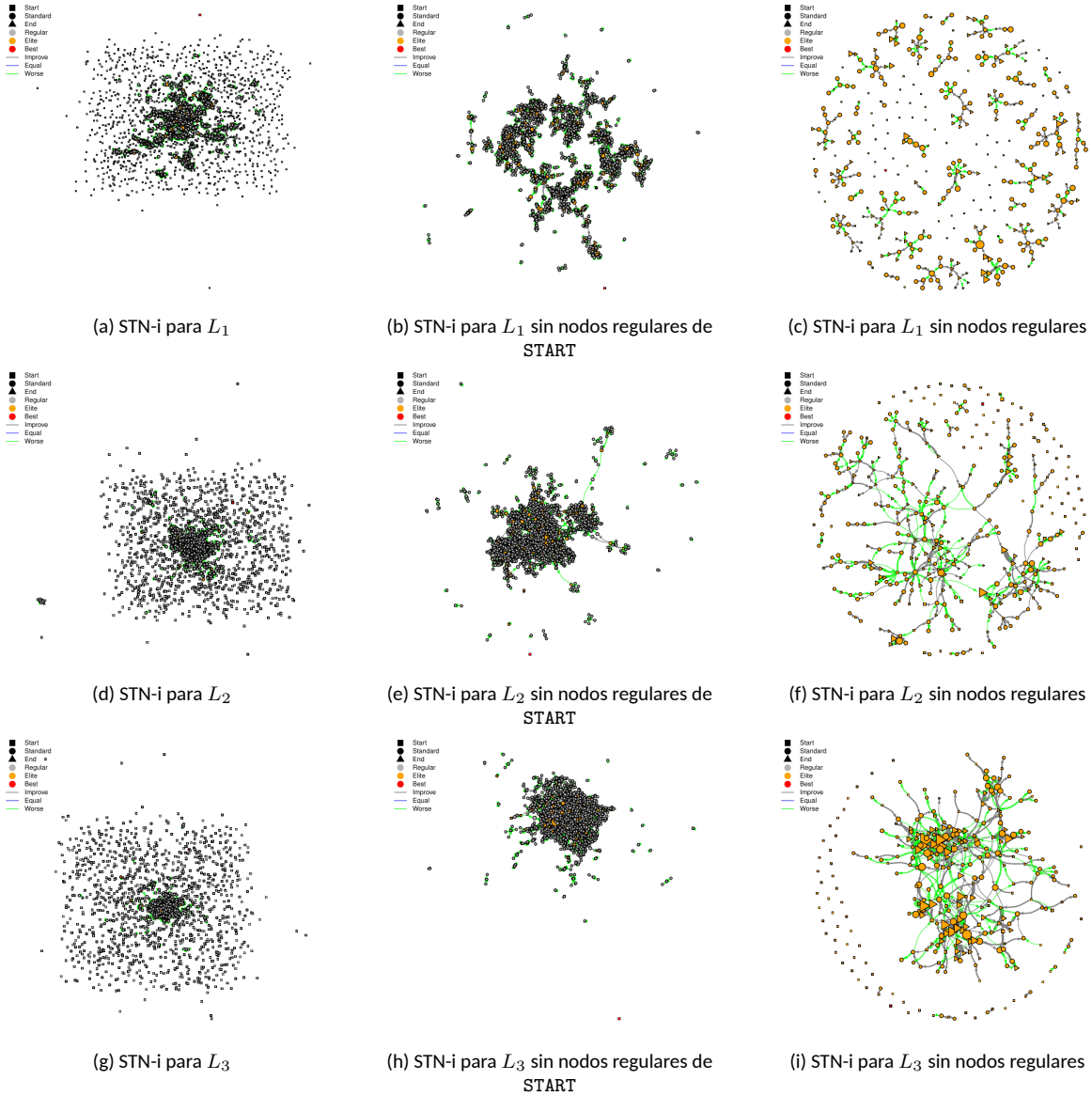


Figura 41: Visualización de las STNs-i generadas a partir del escenario BL-SR para la instancia *PSO-Uni* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

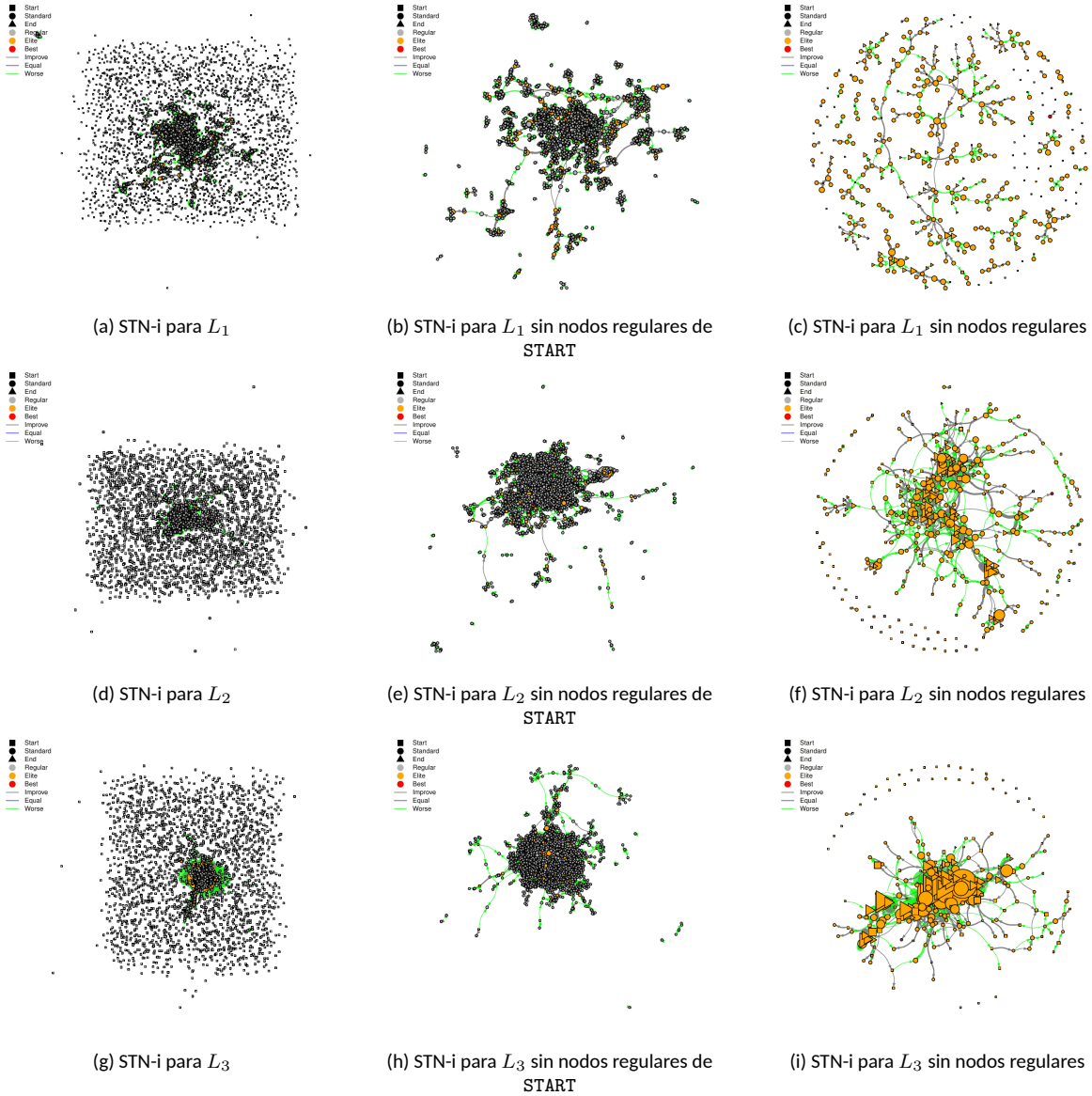


Figura 42: Visualización de las STNs-i generadas a partir del escenario BH-WSR para la instancia *PSO-Uni* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

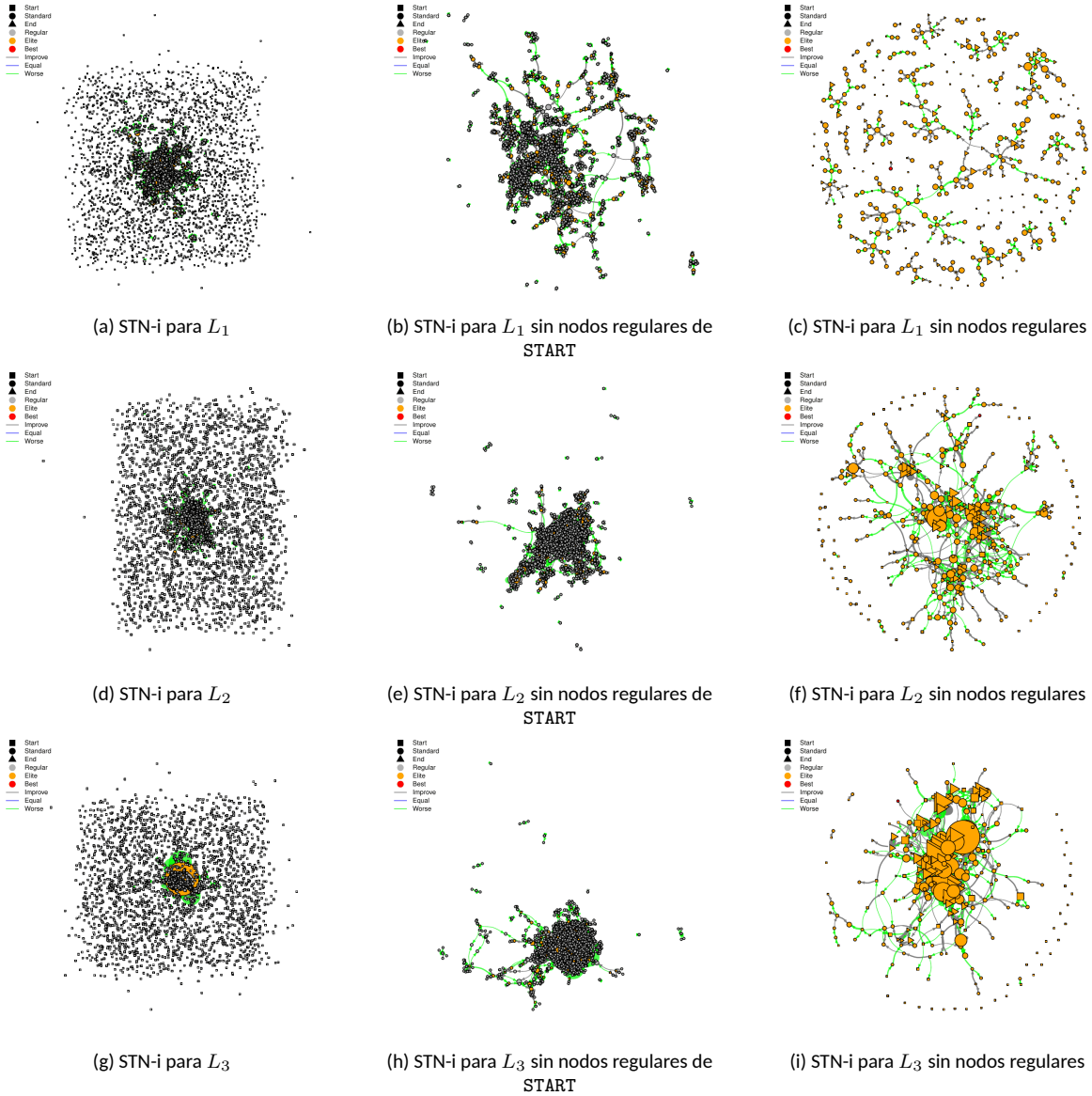


Figura 43: Visualización de las STNs-i generadas a partir del escenario BH-SR para la instancia *PSO-Uni* utilizando el algoritmo PSO-X. Cada fila muestra las redes obtenidas para las locaciones L_1 , L_2 y L_3 , con tres configuraciones distintas: (a, d, g) redes completas, (b, e, h) redes sin nodos regulares provenientes de START, y (c, f, i) redes sin nodos regulares.

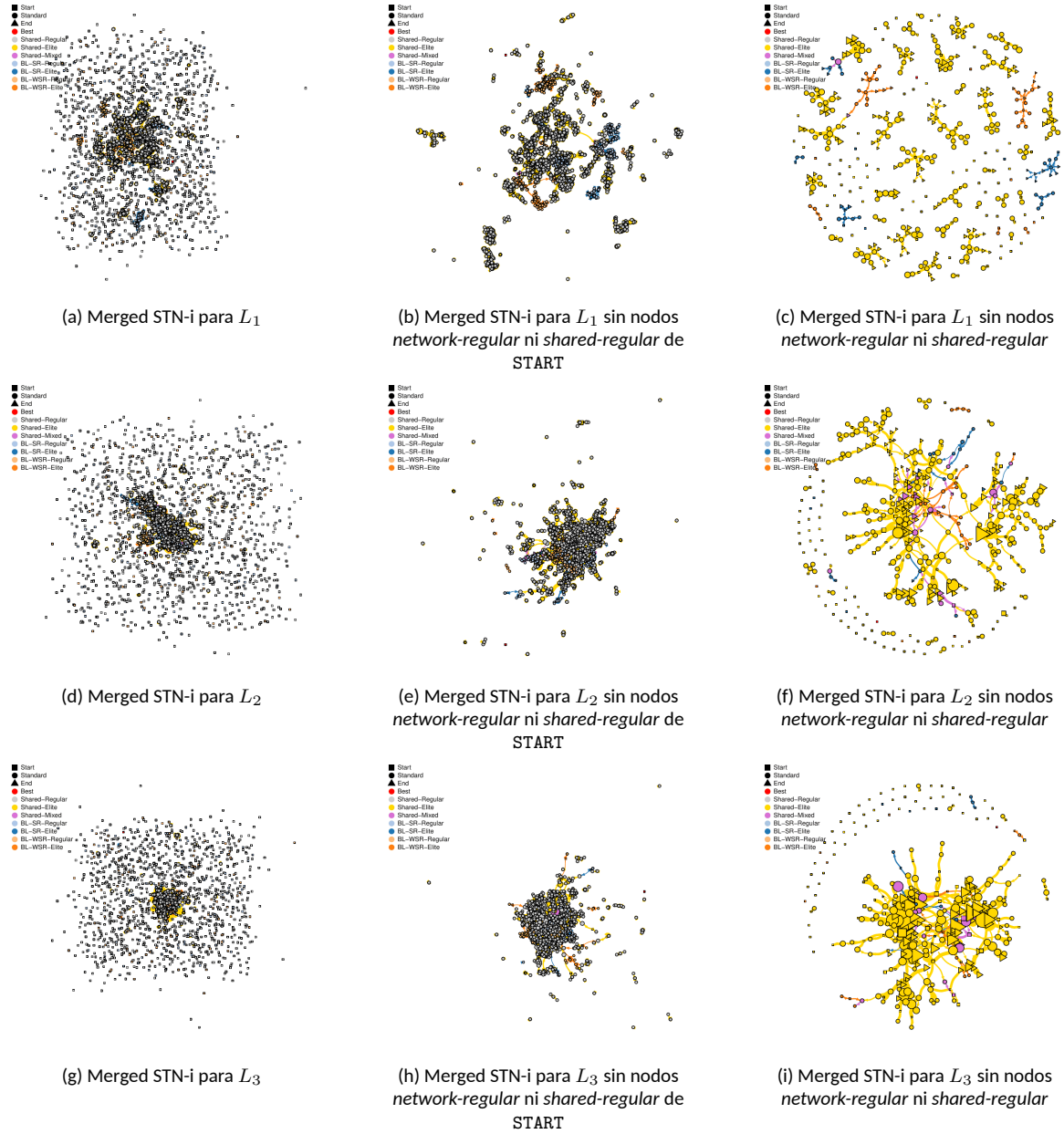


Figura 44: Visualización de las merged STNs-i combinando los escenarios BL-WSR y BL-SR para la instancia *PSO-Uni* con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

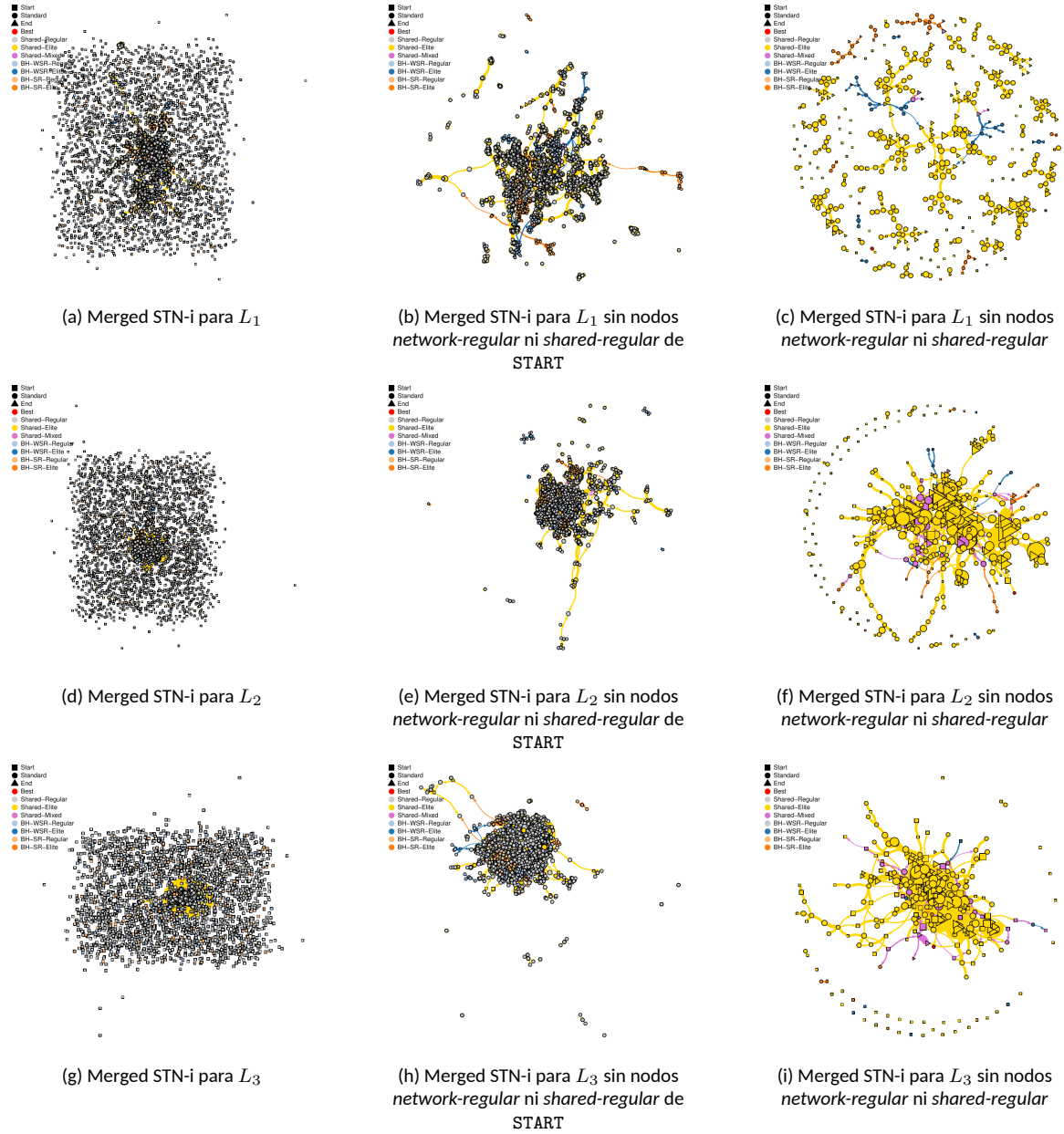


Figura 45: Visualización de las merged STNs-i combinando los escenarios BH-WSR y BH-SR para la instancia *PSO-Uni* con el algoritmo PSO-X. Cada fila representa una locación (L_1 , L_2 , L_3), con tres variantes de red: (a, d, g) redes completas, (b, e, h) redes filtradas sin nodos *network-regular* ni *shared-regular* provenientes de START, y (c, f, i) redes con todos los nodos regulares eliminados.

DISEÑO E IMPLEMENTACIÓN DE MODELO DE SEARCH TRAJECTORY NETWORKS ADAPTADO AL SINTONIZADOR
IRACE

Escenario	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	ed	$tend_{ed}$	$p_{c-n_{re}}$	$p_{c-n_{el}}$	$p_{c-n_{st}^e}$	$p_{c-n_{st}^t}$	$tend_c$	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR	L_1	5121	0.9217	0.0781	0.0002	3811	0.3542	0.8292	0.1708	0.8365	0.1635	0.2761	1311	0.0000	0.0000	1
	L_2	4628	0.9211	0.0787	0.0002	3624	0.3444	0.8292	0.1708	0.8341	0.1659	0.2761	1276	0.0000	0.0000	1
	L_3	3765	0.9201	0.0797	0.0003	3199	0.3357	0.8292	0.1708	0.8443	0.1557	0.2761	1217	0.0000	0.0000	1
BL-SR	L_1	5118	0.9222	0.0776	0.0002	3806	0.3537	0.8292	0.1708	0.8379	0.1621	0.2769	1313	0.0000	0.0000	1
	L_2	4608	0.9210	0.0788	0.0002	3618	0.3513	0.8292	0.1708	0.8356	0.1644	0.2769	1276	0.0000	0.0000	1
	L_3	3751	0.9187	0.0810	0.0003	3172	0.3443	0.8292	0.1708	0.8455	0.1545	0.2769	1224	0.0000	0.0000	1
BH-WSR	L_1	10190	0.9571	0.0428	0.0001	7634	0.3545	0.9028	0.0972	0.9170	0.0830	0.2061	2629	0.0500	1.0000	1
	L_2	7979	0.9530	0.0469	0.0001	6624	0.3536	0.9028	0.0972	0.9165	0.0835	0.2061	2579	0.0500	14.0385	26
	L_3	6264	0.9579	0.0420	0.0002	5239	0.3295	0.9028	0.0972	0.9202	0.0798	0.2061	2430	0.0500	8.2308	39
BH-SR	L_1	10227	0.9568	0.0431	0.0001	7666	0.3428	0.9027	0.0973	0.9183	0.0817	0.2054	2625	0.0500	1.0000	1
	L_2	8027	0.9523	0.0476	0.0001	6577	0.3363	0.9027	0.0973	0.9217	0.0783	0.2054	2580	0.0500	17.5909	22
	L_3	6271	0.9561	0.0437	0.0002	5249	0.3100	0.9027	0.0973	0.9239	0.0761	0.2054	2419	0.0500	8.2564	39

Tabla 19: Métricas extraídas de las STNs-i construidas a partir de las ejecuciones independientes de irace para cada uno de los escenarios del algoritmo PSO-X sobre la instancia *PSO-Uni*.

Escenarios	Locación	n	$p_{n_{re}}$	$p_{n_{el}}$	$p_{n_{bt}}$	$p_{n_{sh}}$	$p_{n_{shre}}$	$p_{n_{shel}}$	$p_{n_{shms}}$	ed	co	$str_{n_{bt}}^{in}$	len_{pt}	pt
BL-WSR $\cup$ BL-SR	L_1	5634	0.1681	0.0144	0.0002	0.8174	0.7536	0.0634	0.0004	4194	1441	0.0000	0.0000	1
	L_2	5049	0.1608	0.0097	0.0002	0.8293	0.7602	0.0652	0.0040	3995	1401	0.0000	0.0000	1
	L_3	4075	0.1487	0.0066	0.0002	0.8444	0.7718	0.0690	0.0037	3511	1334	0.0000	0.0000	1
BH-WSR $\cup$ BH-SR	L_1	11220	0.1733	0.0070	0.0001	0.8197	0.7839	0.0354	0.0004	8417	2886	0.0500	2.0000	1
	L_2	8741	0.1647	0.0041	0.0001	0.8311	0.7881	0.0395	0.0034	7261	2835	0.0500	23.4643	28
	L_3	6797	0.1536	0.0022	0.0001	0.8442	0.8037	0.0365	0.0038	5755	2649	0.0500	13.1628	43

Tabla 20: Métricas de las STNs-i resultantes de la fusión de los escenarios BL-WSR con BL-SR, y BH-WSR con BH-SR, para el algoritmo PSO-X en la instancia *PSO-Uni*.

Escenario	Configuraciones	Iter. Mín	Iter. Máx	Iter. Prom	Uso de <i>soft-restart</i>
BL-WSR	5130	6	9	6.9500	–
BL-SR	5126	6	9	7.0000	0.0000
BH-WSR	10343	6	9	7.4000	–
BH-SR	10394	6	9	7.4500	0.0000

Tabla 21: Resumen de datos globales de irace sobre las ejecuciones independientes de los escenarios considerados para la instancia *PSO-Uni* del algoritmo PSO-X.

4.2.10. Análisis de PSO-X (PSO-Uni)

El conjunto *PSO-Uni* se compone de funciones unimodales provenientes de las competencias CEC'05, CEC'14 y SOCO'10, todas definidas en un espacio de 30 dimensiones. A diferencia de los casos híbrido o multimodal, estas funciones presentan un único óptimo global y estructuras más suaves, sin grandes saltos ni múltiples valles. Sin embargo, el proceso de sintonización de PSO-X no se ve considerablemente simplificado, ya que el espacio de parámetros sigue siendo altamente combinatorio y sensible, con interacciones complejas entre componentes de la topología y mecanismos de actualización del algoritmo.

Este conjunto actúa como línea base estructural frente a los dominios más complejos de *PSO-Mix* y *PSO-Mul*, permitiendo evaluar si la reducción en la complejidad del problema objetivo se traduce en una mejora estructural en las redes STN-i construidas. Dado el tipo de funciones, se esperaría una menor dispersión en la red, con trayectorias algo más dirigidas y menos fragmentación, pero sin eliminar por completo los efectos del espacio paramétrico denso.

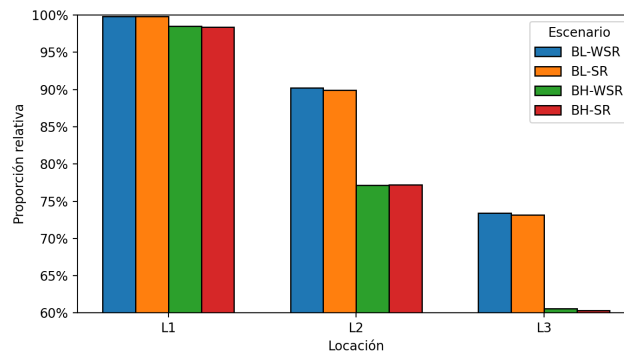


Figura 46: Proporción relativa de locaciones sobre el total de configuraciones de cada escenario para la instancia *PSO-Uni* del algoritmo PSO-X.

La Figura 46 muestra un patrón consistente con los otros dominios de PSO-X: las locaciones L_1 y L_2 concentran más del 90 % de las configuraciones, mientras que L_3 reduce esta proporción al rango del 60–75 %. Esta caída más pronunciada en L_3 respecto a *PSO-Mul* evidencia una mejor capacidad de síntesis estructural en discretizaciones más finas, sugiriendo que las regiones relevantes del espacio pueden ser representadas con mayor compacidad cuando las trayectorias son menos dispersas, como es el caso unimodal.

La Tabla 21 revela que el número de iteraciones promedio es más alto que en los otros dominios, alcanzando valores entre 6.95 y 7.45. Este comportamiento sugiere que, en este caso, *irace* logra mantener vivas más configuraciones por iteración, lo cual implica una menor tasa de descarte por ronda. Esto puede deberse a una menor variabilidad entre configuraciones evaluadas, típica de dominios con trayectorias más consistentes. A pesar de ello, el mecanismo *soft-restart* no se activa en ningún escenario, lo cual refuerza la idea de que el proceso mantiene diversidad suficiente para no gatillar condiciones de reinicio.

Las Figuras 40 a 43 muestran una vez más la presencia masiva de nodos descartados en etapas iniciales. Sin embargo, en las vistas filtradas se observa una mayor desconexión que en *PSO-Mix* y *PSO-Mul*, con fragmentos más separados y una estructura menos continua, especialmente en granularidades altas. A pesar de ello, se identifican subredes élite mejor conectadas en escenarios de alto presupuesto, lo cual indica que *irace* logra descubrir rutas más estables hacia configuraciones de calidad cuando dispone de mayor presupuesto.

Desde las métricas (Tabla 19), se constata que el número de arcos es inferior al de nodos en todos los casos, y el valor de $tend_{ed}$ alcanza niveles mínimos (≈ 0.33), lo cual indica que una proporción considerable de las transiciones entre configuraciones ocurre con degradación de calidad relativa. Esto sugiere que el espacio, aunque más estructurado, aún está dominado por trayectorias con bajo rendimiento.

En todos los escenarios se mantiene una proporción elevada de nodos regulares (91–96 %), incluso superior a *PSO-Mul* (Tabla 19). Esto refuerza la interpretación de que la mayoría de las configuraciones son descartadas tempranamente. Las configuraciones élite se concentran mayoritariamente en nodos no iniciales, y su proporción es ligeramente superior que en dominios anteriores. Esta distribución homogénea y recurrente a lo largo del tiempo refleja un comportamiento más estable de *irace* en este dominio.

En los escenarios de bajo presupuesto, los mejores nodos coinciden con nodos de inicio ($str_{nbt}^{in} = 0$), sin trayectorias asociadas, lo que indica una falta de exploración estructurada hacia ellos. En cambio, en los escenarios de alto presupuesto, se detectan trayectorias más largas y numerosas (hasta 17 caminos), reflejando una conectividad creciente hacia regiones de alto rendimiento. Esto es indicativo de una convergencia estructural más articulada, impulsada por una mayor cobertura del espacio de búsqueda, como se observa también en la subfigura (i) de la Figura 43.

La métrica $tend_c$ alcanza sus valores más altos en este dominio, llegando a 0.276 en escenarios de bajo presupuesto (Tabla 19). Esto sugiere que las configuraciones élite tienden a mantenerse activas por más iteraciones, lo cual puede explicarse por la menor variabilidad del dominio y la estabilidad relativa de las trayectorias generadas. En este sentido, *irace* parece beneficiarse de un entorno más predecible, donde la presión selectiva no fuerza un descarte tan acelerado como en dominios más complejos.

Las redes fusionadas (Figuras 44 y 45) mantienen una estructura muy densa en nodos descartados de inicio. Una vez filtrados, se aprecian agrupamientos de mayor cohesión estructural, pero aún con múltiples componentes disconexas. A diferencia de dominios anteriores, aquí aparecen con más claridad nodos particulares de cada ejecución (no compartidos), incluso en locaciones más finas. Esto confirma que, aunque las trayectorias son más consistentes, la diversidad de soluciones exploradas sigue siendo alta.

Las proporciones de nodos compartidos (p_{nsh}) superan el 81 % en todos los escenarios (Tabla 20), manteniéndose ligeramente por debajo de *PSO-Mix*, pero aún reflejando un alto grado de coincidencia estructural. La mayoría de los nodos compartidos siguen siendo regu-

lares ($p_{n_{shre}} > 75\%$), aunque los nodos mixtos aparecen incluso en L_1 , lo cual sugiere que algunas configuraciones tienen una representación ambigua entre ejecuciones. Los nodos élite compartidos se mantienen en proporciones reducidas pero consistentes, reforzando la interpretación de que las regiones de alto rendimiento son exploradas de forma más localizada y heterogénea.

Las métricas asociadas a los mejores nodos mantienen el patrón de los escenarios individuales. En particular, en escenarios de alto presupuesto, los caminos hacia estos nodos son más numerosos y largos, alcanzando hasta 17 pasos (Tabla 20). Esta mejora se explica por la fusión de ejecuciones, que permite consolidar caminos dispersos y convertir nodos de inicio en intermediarios o de fin. El resultado es una estructura más rica en accesos hacia regiones élite, sin perder la diversidad que caracteriza este dominio.

4.2.11. Análisis general

El análisis desarrollado a partir de la aplicación del modelo STN-i sobre los diferentes dominios —MMASQAP en QAP-60, ACOTSP en TSP-2000, y PSO-X en las familias de funciones *PSO-Mix*, *PSO-Mul* y *PSO-Uni*— ha permitido caracterizar de forma estructurada y comparativa el comportamiento de *irace* frente a distintos niveles de complejidad, tanto desde la perspectiva paramétrica como desde las propiedades funcionales de los problemas objetivo.

Desde el punto de vista de los algoritmos considerados, se observan diferencias marcadas en el diseño y en la dificultad estructural del espacio de búsqueda generado. MMASQAP, con tres parámetros discretos y sin dependencias condicionales, representa un caso de baja dimensionalidad y estructura simple. En este escenario, las STN-i generadas tienden a mostrar conectividad elevada, fragmentación reducida y una alta proporción de nodos compartidos, especialmente en locaciones más gruesas (como L_3). El proceso de sintonización sobre este dominio favorece la repetición de trayectorias y una rápida identificación de regiones favorables, lo que se refleja en métricas de alta coherencia estructural y supervivencia sostenida de configuraciones.

En contraste, ACOTSP presenta un espacio paramétrico más amplio (11 parámetros sintonizados, de los cuales 8 fueron discretizados), incluyendo múltiples dependencias condicionales y codificaciones mixtas. Esta complejidad estructural sumado a las interacciones entre los parámetros introducen una elevada dispersión en las configuraciones exploradas, dificultando la consolidación de trayectorias y generando STN-i fragmentadas, con menor conectividad entre nodos relevantes. La presencia de nodos élite compartidos es escasa, y la direccionalidad de las transiciones suele ser baja, indicando un comportamiento exploratorio menos estructurado que en MMASQAP.

PSO-X representa el caso más complejo, tanto por el diseño del algoritmo como por los dominios de funciones utilizados. Aunque solo 6 de los 8 parámetros fueron discretizados, estos incluyen componentes condicionales y categóricos de amplio rango, lo que genera un espacio altamente combinatorio y difícil de representar estructuralmente. Esta complejidad se ve intensificada al considerar que el proceso de sintonización se aplica sobre tres conjuntos de funciones de distinta naturaleza: *PSO-Mix* (híbridas y de composición), *PSO-Mul* (altamente multimodales), y *PSO-Uni* (unimodales). En los dos primeros, las STN-i muestran una dispersión extrema, fragmentación elevada y escasa conectividad hacia nodos de alta calidad, incluso bajo escenarios de alto presupuesto. Las métricas estructurales reflejan una baja supervivencia de configuraciones élite, trayectorias poco articuladas y presencia mayoritaria de nodos descartados en etapas tempranas. El caso *PSO-Uni*, en cambio, permite observar una mejora relativa en la conectividad de nodos élite y en la existencia de trayectorias largas hacia configuraciones destacadas, producto de una menor complejidad funcional en las funciones objetivo.

En este marco, las STN-i permiten evidenciar cómo la dificultad del proceso de sintonización no depende exclusivamente del número de parámetros involucrados, sino también de su

tipo, interdependencia y sensibilidad frente al problema objetivo. Del mismo modo, se observa que la naturaleza del conjunto de instancias influye significativamente en la forma de las redes generadas: mientras que dominios homogéneos como QAP-60 o TSP-2000 inducen trayectorias más repetibles y estructuras más cohesionadas, los dominios continuos con alta variabilidad interna —como PSO-Mix y PSO-Mul— promueven una exploración más diversa y dispersa.

Este efecto no solo se refleja en la estructura de las redes, sino también en el comportamiento dinámico de *irace* durante la sintonización. En particular, se evidencia en el uso diferenciado de mecanismos internos como el *soft-restart*. Este mecanismo fue activado únicamente en el dominio MMASQAP, reflejo de la baja dimensionalidad del espacio de búsqueda y su propensión a converger rápidamente hacia configuraciones subóptimas. La activación del *soft-restart* implica el reinicio parcial del proceso, permitiendo a *irace* explorar nuevas regiones del espacio con un conjunto renovado de configuraciones iniciales. En cambio, en dominios más amplios y heterogéneos como ACOTSP o PSO-X, la exploración permanece activa por más tiempo, y no se alcanzan las condiciones necesarias para activar dicho mecanismo. Esto evidencia que la estructura del espacio paramétrico no solo afecta la forma de las redes STN-i, sino también las dinámicas internas de exploración y diversificación dentro del proceso de sintonización automática.

Estos contrastes se representan de forma sintética en la Figura 47, donde se muestra la proporción promedio entre el número de nodos generados y la cantidad total de configuraciones evaluadas por *irace* en cada red STN-i, agrupando por combinación *algoritmo-conjunto de instancias* y diferenciando los escenarios de bajo y alto presupuesto. Esta medida permite estimar indirectamente el nivel de agrupamiento o fragmentación estructural en las redes, ya que valores altos indican una escasa coincidencia entre configuraciones (es decir, alta fragmentación), mientras que proporciones más bajas reflejan una mayor capacidad del modelo STN-i para consolidar configuraciones similares en una misma locación. Por ejemplo, MMASQAP sobre QAP-60 presenta una síntesis estructural elevada, con proporciones consistentemente bajas incluso bajo discretizaciones finas, mientras que dominios como PSO-Mix o PSO-Mul alcanzan valores cercanos al 100 %, evidenciando una baja capacidad de agrupamiento incluso en escenarios de alta granularidad.

Para complementar esta caracterización, la Figura 48 presenta un análisis más detallado de las métricas estructurales clave derivadas de las STN-i, utilizando un esquema de radar agrupado por algoritmo y diferenciando los escenarios de bajo (BL) y alto (BH) presupuesto. En (a), correspondiente a MMASQAP sobre QAP-60, se observa una fuerte predominancia de nodos regulares ($p_{n_{re}}$), así como una alta proporción de configuraciones iniciales concentradas en estos ($p_{c-n_{re}^{st}}$), con baja presencia de configuraciones élite en etapas tempranas. Las diferencias entre presupuestos son sutiles, lo que evidencia la consistencia estructural del proceso de sintonización en este dominio.

En (b), correspondiente a ACOTSP sobre TSP-2000, se mantiene la tendencia hacia nodos regulares, pero con una mayor proporción de configuraciones élite en nodos de inicio, especialmente bajo presupuesto bajo. Las métricas $tend_{ed}$ y $tend_c$ reflejan una menor direc-

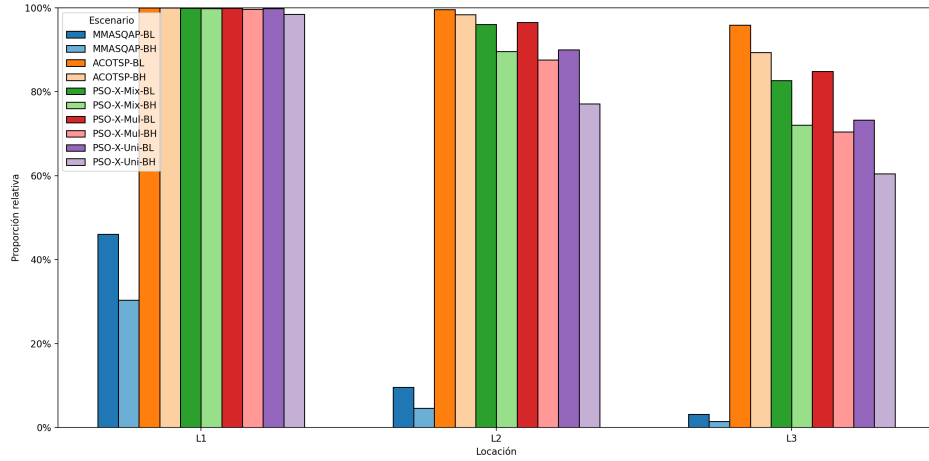


Figura 47: Proporción promedio entre nodos y configuraciones en las STN-i, diferenciando por combinación de algoritmo objetivo, conjunto de instancias y nivel de presupuesto.

cionalidad y supervivencia que en MMASQAP, indicando que el espacio de búsqueda más amplio y condicional impacta en la estabilidad del proceso.

Por último, en (c), que agrupa los escenarios de PSO-X en sus tres dominios funcionales, se aprecian diferencias más marcadas entre funciones. Las funciones *PSO-Mix* y *PSO-Mul* presentan menor direccionalidad y supervivencia de configuraciones, con una dispersión más alta en las métricas relativas. En contraste, *PSO-Uni* exhibe una estructura más estable, con una mayor consistencia entre niveles de presupuesto y valores superiores en conectividad hacia nodos relevantes. Este conjunto de gráficos refuerza la interpretación de que la estructura del espacio funcional tiene un impacto más pronunciado que el presupuesto computacional asignado.

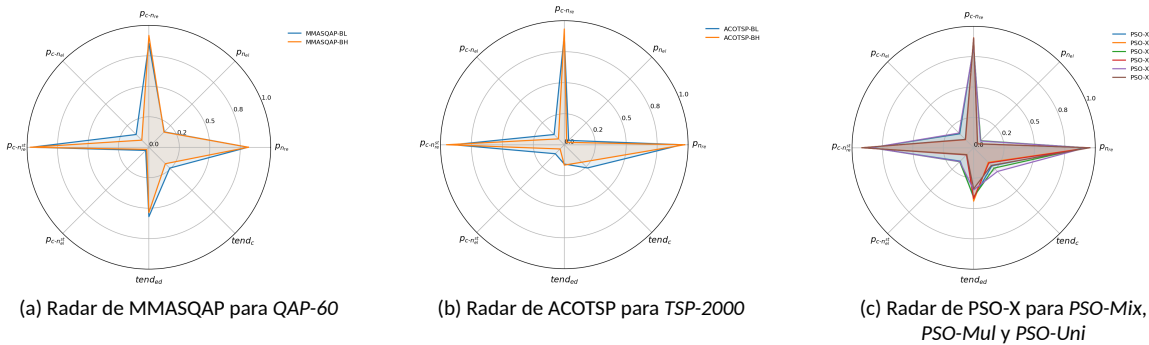


Figura 48: Representación de las tendencias de las STN-i para las métricas más relevantes, agrupadas por cada nivel de presupuesto computacional.

De forma complementaria, la Figura 49 analiza las redes merged STN-i, enfocándose en la estructura compartida entre ejecuciones. En (a), correspondiente nuevamente a MMASQAP,

se aprecia una proporción extremadamente alta de nodos compartidos ($p_{n_{sh}}$), con predominancia de nodos regulares compartidos ($p_{n_{shre}}$), lo que confirma la estabilidad exploratoria observada en los escenarios individuales.

En (b), ACOTSP presenta proporciones significativamente menores de nodos compartidos, con una mayor dispersión entre ejecuciones y escasa presencia de nodos mixtos o élite compartidos. Esto refuerza la hipótesis de que la estructura jerárquica y la condición de los parámetros reducen la convergencia estructural en este dominio.

Finalmente, (c) presenta los resultados para PSO-X en sus tres dominios, destacando que en *PSO-Mix* bajo alto presupuesto se alcanza un mayor grado de coincidencia estructural ($p_{n_{sh}}$ cercano a 1), mientras que *PSO-Mul* y *PSO-Uni* presentan distribuciones más equilibradas y dominadas por nodos regulares compartidos. A pesar de las diferencias entre dominios, las tendencias entre presupuestos dentro de cada dominio tienden a mantenerse, lo cual sugiere que el presupuesto tiene un efecto menos relevante que la naturaleza del problema en la formación de redes estables.

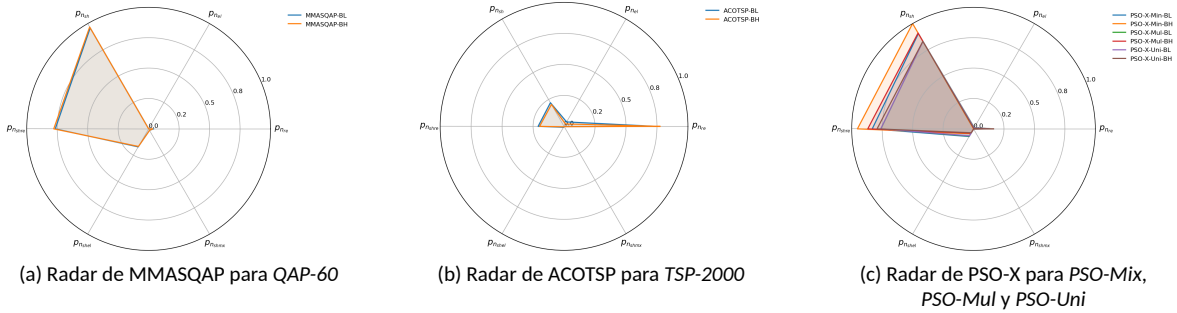


Figura 49: Representación de las tendencias de las merged STN-i para las métricas más relevantes, agrupadas por cada nivel de presupuesto computacional.

La estructura de las visualizaciones de las STN-i y merged STN-i presentadas en este trabajo está fuertemente influida por decisiones técnicas adoptadas en su modelado. En particular, el uso sistemático del algoritmo de disposición Fruchterman-Reingold (FR) permitió generar representaciones que acentúan la conectividad entre nodos y la organización espacial de las trayectorias. Esta disposición facilitó la identificación visual de patrones relevantes como núcleos de convergencia, trayectorias dominantes y subcomponentes desconectados, especialmente útiles en dominios estructuralmente simples como MMASQAP o más fragmentados como *PSO-Mul*. Así, el layout FR no solo mejora la legibilidad de las redes, sino que condiciona directamente la forma en que se interpretan sus dinámicas internas.

Otro elemento clave en este análisis y en los resultados obtenidos es la métrica seleccionada para representar la calidad de los nodos. En esta investigación se utilizó la media de las calidades de las configuraciones agrupadas, lo cual ofrece una visión global del rendimiento local. No obstante, esta elección puede ser sensible a valores atípicos, afectando la asignación funcional de los nodos en la red. Este efecto se ve amplificado por la jerarquía de etiquetado

entre nodos de tipo START, END y BEST, donde ciertos nodos pueden aparecer como relevantes sin estar topológicamente conectados o mostrar trayectorias nulas ($len_{pt} = 0$), o bien quedar desplazados por decisiones de clasificación. Esto impacta tanto en la percepción de regiones de alto rendimiento como en la coherencia estructural de la red.

En suma, la forma y semántica de las STN-i y merged STN-i no reflejan únicamente el comportamiento exploratorio de irace, sino también una cadena de decisiones técnicas que definen su construcción y representación. Estas incluyen desde elementos de modelado como la activación del mecanismo de *soft-restart* —presente solo en dominios simples como MMASQAP— hasta parámetros como la granularidad de discretización, la codificación de los hiperparámetros y la forma de visualizar y etiquetar nodos. Comprender la interacción entre estos factores resulta esencial para interpretar de forma rigurosa los patrones de exploración, convergencia y estabilidad en procesos de sintonización automática.

4.3. Conclusiones

La presente sección ha permitido caracterizar en profundidad el comportamiento exploratorio de irace a través del modelo STN-i, bajo un conjunto amplio de condiciones experimentales que combinan algoritmos objetivo de distinta naturaleza (MMASQAP, ACOTSP y PSO-X), conjuntos de instancias representativos, y tres niveles de discretización paramétrica (locaciones L_1 , L_2 y L_3). Esta configuración experimental, complementada con la evaluación en dos niveles de presupuesto computacional (bajo y alto), ha permitido explorar de forma comparativa cómo influyen la complejidad del espacio de búsqueda y las decisiones de representación sobre las trayectorias generadas por el proceso de sintonización.

Los resultados muestran que dominios de baja complejidad estructural y funcional, como MMASQAP sobre QAP-60, producen STN-i altamente cohesivas, con trayectorias bien definidas, nodos repetidos entre ejecuciones y una alta capacidad de síntesis incluso en locaciones finas. En estos casos, se observó además la activación efectiva del mecanismo de *soft-restart*, evidenciando una rápida convergencia local que requiere diversificación para continuar la exploración. En contraste, dominios con mayor complejidad, como PSO-X sobre PSO-Mix o PSO-Mul, muestran una exploración mucho más dispersa, STN-i fragmentadas, menor coincidencia estructural y una clara dificultad para consolidar regiones de alta calidad, incluso con mayor presupuesto. ACOTSP, por su parte, con una alta dimensionalidad y presencia de parámetros condicionales, presenta un comportamiento intermedio, sensible tanto al presupuesto como a la granularidad utilizada.

En términos de visualización, las STN-i y merged STN-i demostraron ser herramientas expresivas para evidenciar patrones estructurales y dinámicas exploratorias. El uso del layout Fruchterman-Reingold permitió resaltar zonas de convergencia, trayectorias dominantes, regiones desconectadas y nodos relevantes desde un punto de vista visual. La inclusión de métricas estructurales, tanto absolutas como proporcionales, facilitó la comparación entre dominios y presupuestos, permitiendo observar diferencias de comportamiento que no se-

rían evidentes únicamente desde los resultados numéricos. Las merged STN-i, en particular, permitieron cuantificar la estabilidad de las exploraciones a través del análisis de nodos compartidos y su tipo, lo cual resultó clave para evaluar la coherencia estructural de irace bajo diferentes condiciones.

En conjunto, esta sección refuerza el valor de las STN-i como herramienta de análisis estructural del comportamiento de irace. Sin embargo, también evidencia que su uso efectivo requiere un conocimiento profundo no solo del dominio y del algoritmo sintonizado, sino también del funcionamiento interno del proceso de sintonización, incluyendo componentes como el *soft-restart*, así como de los criterios de modelado empleados: discretización, agrupamiento, etiquetado, métrica de calidad y algoritmo de disposición gráfica. La estructura resultante no es una representación neutra, sino que depende de todas estas decisiones, las cuales condicionan directamente la lectura, la interpretación y la utilidad analítica de las redes generadas. Por ello, la aplicación de STN-i exige una mirada crítica e informada que articule tanto la semántica visual como el conocimiento técnico y contextual del problema abordado.

CAPÍTULO 5

CONCLUSIONES

El problema de configuración de algoritmos es fundamental en el ámbito de la optimización moderna, especialmente frente al auge de algoritmos metaheurísticos aplicables a dominios complejos y altamente variables. Dentro de las estrategias existentes para enfrentar este desafío, la configuración automática de parámetros se ha consolidado como una solución flexible y eficaz, permitiendo ajustar algoritmos sin necesidad de modificar sus componentes internos.

Entre las herramientas más representativas en este dominio, *irace* destaca por su enfoque basado en *racing*, integrando criterios estadísticos para guiar el proceso de selección de configuraciones prometedoras. Sin embargo, su naturaleza iterativa, el uso de múltiples configuraciones en paralelo y la aplicación de estrategias como el *soft-restart* generan una dinámica compleja, poco transparente desde una perspectiva externa. Esta opacidad limita la comprensión profunda de cómo se estructura y evoluciona la búsqueda durante el proceso de sintonización.

En respuesta a esta problemática, se desarrolló el modelo STN-i, una adaptación de las Search Trajectory Networks (STNs) al contexto específico del sintonizador *irace*. Esta propuesta se fundamenta en la necesidad de capturar las dinámicas propias del proceso de sintonización basado en *racing*, donde múltiples configuraciones son evaluadas y generadas en paralelo a partir de un conjunto de configuraciones élite por iteración, lo cual rompe con la linealidad secuencial que asumían las STNs tradicionales.

Para ello, STN-i introduce una estructura de trayectorias paralelas mediante el concepto de *parent-edges*, que conectan directamente a cada configuración generada con su(s) configuración(es) élite responsable(s) en la iteración correspondiente. A su vez, se incorpora una taxonomía extendida de nodos, donde cada nodo representa una locación en el espacio discretizado de configuraciones. Estos nodos se clasifican tanto a nivel topológico en la red (como START, STANDARD y END) como a nivel estructural, diferenciando entre configuraciones regulares, élites o aquellas de mejor calidad global.

La consolidación de esta información permite capturar con precisión tanto la evolución de las configuraciones como su relevancia dentro de la estructura de búsqueda, diferenciando, por ejemplo, si una configuración fue considerada élite de forma local pero descartada a nivel global. Esta granularidad semántica resulta esencial para comprender fenómenos como la concentración excesiva en regiones específicas del espacio, la convergencia prematura o la exploración divergente de configuraciones.

Adicionalmente, se definió la variante merged STN-i, la cual fusiona múltiples STN-i generadas a partir de ejecuciones independientes del mismo escenario experimental (por ejemplo, con o sin *soft-restart*, o con distintos niveles de presupuesto). Este modelo incorpora meca-

nismos especiales de consolidación, manteniendo la clasificación topológica equivalente a las STN-i individuales, pero introduciendo una clasificación estructural más rica. En particular, se identifican los nodos compartidos como *shared-regular*, *shared-elite* y *shared-mixed*, los cuales reflejan diferentes niveles de coincidencia entre ejecuciones. A su vez, se conservan las categorías *network-regular* y *network-elite* para aquellos nodos que pertenecen exclusivamente a una de las redes fusionadas. Este diseño permite no solo visualizar y analizar regiones del espacio de búsqueda exploradas en común, sino también identificar diferencias estructurales o convergencias hacia zonas específicas, lo cual resulta fundamental para contrastar comportamientos bajo distintas políticas de sintonización o configuraciones experimentales.

El resultado es un modelo generalizado, con un fuerte sustento en la semántica operativa de *irace*, capaz de representar con fidelidad trayectorias complejas, simultáneas y no lineales. Gracias a esta capacidad, STN-i y merged STN-i se posicionan como herramientas analíticas robustas, adecuadas tanto para el estudio detallado del comportamiento exploratorio en procesos de sintonización automática, como para su futura extensión a contextos más generales en la optimización basada en trayectorias.

La validación empírica del modelo propuesto se llevó a cabo mediante un conjunto extenso y controlado de escenarios de sintonización, con el objetivo de evaluar su capacidad descriptiva, analítica y comparativa. Para ello, se consideraron tres algoritmos metaheurísticos (MMASQAP, ACOTSP y PSO-X), cada uno sintonizado sobre un conjunto específico de instancias (QAP-60, TSP-2000, PSO-Mix, PSO-Mul y PSO-Uni, respectivamente). Cada configuración experimental fue evaluada bajo dos niveles de presupuesto computacional y con o sin la activación del mecanismo de *soft-restart*, conformando un total de 20 combinaciones experimentales distintas. Para cada una de ellas se realizaron 20 ejecuciones independientes mediante *irace*, manteniendo iguales las semillas de aleatoriedad en los pares de ejecuciones con igual presupuesto dentro de cada grupo algoritmo-conjunto de instancias, generando así un conjunto de datos de sintonización estructurado y replicable.

A partir de estos datos, se construyeron las redes STN-i correspondientes aplicando, para cada combinación experimental, tres discretizaciones diferentes del espacio de parámetros: L_1 , L_2 y L_3 , representando granularidades fina, media y gruesa, respectivamente. Estas discretizaciones se aplicaron posterior a la ejecución de *irace*, como parte del proceso de construcción de las STN-i, con el propósito de analizar cómo la resolución espacial de las locaciones afecta la estructura de las redes y la interpretación de las trayectorias de búsqueda. En total, se generaron 60 redes STN-i, de las cuales se construyeron 30 redes merged STN-i, agrupando pares de ejecuciones que compartían el mismo presupuesto computacional y el mismo par algoritmo-conjunto de instancias, diferenciándose únicamente por la activación o no del mecanismo de *soft-restart*. Estas agrupaciones se realizaron bajo la misma discretización para cada par, lo que permitió analizar de manera controlada las diferencias estructurales, convergencias y divergencias entre trayectorias generadas bajo condiciones similares pero con políticas de sintonización contrastantes.

Los resultados obtenidos a partir de estas redes permitieron caracterizar, el comportamiento

exploratorio de *irace* en función del tipo de algoritmo, la estructura del espacio de configuración y la influencia de los factores experimentales. Por ejemplo, en el caso de MMAS-QAP, las STN-i mostraron una alta cohesión estructural, con trayectorias recurrentes hacia regiones élite y una conectividad elevada entre nodos relevantes. Este patrón fue más evidente en las discretizaciones gruesas (L_3), donde la agrupación de configuraciones facilitó la identificación de rutas dominantes y regiones de atracción. En contraste, el algoritmo PSO-X presentó redes con una mayor fragmentación, elevada proporción de nodos descartados tempranamente y baja conectividad estructural, especialmente en las discretizaciones finas (L_1), donde la alta resolución evidenció la dispersión de las trayectorias y la falta de convergencia clara. Por su parte, ACOTSP exhibió un comportamiento intermedio, con regiones densa y consistentemente exploradas, pero también zonas de búsqueda más dispersa, influenciadas por el conjunto de instancias y el presupuesto asignado.

Estos patrones fueron cuantificados mediante un conjunto de métricas diseñadas específicamente para el análisis estructural de las STN-i y merged STN-i. Entre ellas se encuentran el número total de nodos n , las proporciones de nodos regulares $p_{n_{re}}$, élite $p_{n_{el}}$ y de mejor calidad $p_{n_{bt}}$, la fuerza de conexión entrante de los mejores nodos $str_{n_{bt}}^{in}$, la longitud promedio de los caminos más cortos a mejores nodos len_{pt} y la cantidad de caminos más cortos pt , entre otras. Estas métricas fueron aplicadas exclusivamente sobre las versiones completas de las redes, ya que constituyen una representación fiel del proceso completo de exploración de *irace* en cada escenario experimental.

El análisis comparativo entre distintos niveles de discretización permitió identificar cómo la granularidad del espacio discretizado afecta directamente la densidad topológica de las redes, la estabilidad de las trayectorias dominantes y la visibilidad de patrones estructurales relevantes. Por ejemplo, en discretizaciones gruesas se observaron redes más compactas y con menor número de trayectorias divergentes, mientras que en discretizaciones finas se evidenció una mayor dispersión estructural y diversidad de rutas.

Además de las métricas cuantitativas, se exploró la posibilidad de representar diferentes vistas o estados de la red, como aquellas sin nodos regulares iniciales o sin ningún nodo regular. Estas representaciones, aunque no fueron empleadas directamente en la medición estructural, resultaron especialmente útiles para la visualización y la interpretación cualitativa de la dinámica del proceso. En particular, permitieron aislar regiones informativas y minimizar la interferencia visual de nodos descartados tempranamente, facilitando el análisis de patrones de convergencia, bifurcaciones o estancamientos exploratorios.

Las visualizaciones generadas mediante el algoritmo de disposición Fruchterman-Reingold fueron fundamentales para revelar la morfología general de las trayectorias, la centralidad estructural de ciertos nodos élite y la existencia de regiones compartidas entre ejecuciones. Este *layout* permitió una representación clara de los patrones de conexión y agrupamiento en las STN-i y merged STN-i, favoreciendo el análisis cualitativo de aspectos como la densidad local, la dispersión de rutas y la predominancia de trayectorias élite. La elección de este algoritmo de disposición se justificó por su capacidad de balancear estructura y legibilidad en redes de tamaño y complejidad variables.

Un aspecto particularmente revelador fue el análisis del comportamiento del mecanismo de *soft-restart*. Su activación se observó principalmente en el caso de MMASQAP, asociado a una convergencia acelerada hacia regiones subóptimas del espacio de parámetros y a la necesidad de reinserción de diversidad. En cambio, en algoritmos como PSO-X y ACOTSP, este mecanismo prácticamente no fue activado, lo cual puede atribuirse a una dinámica exploratoria más sostenida, promovida por la complejidad y dimensionalidad de sus espacios de parámetros.

La construcción y análisis de las redes merged STN-i permitió estudiar comparativamente estas dinámicas, revelando zonas de convergencia estructural mediante la identificación de nodos *shared*, y caracterizando trayectorias exclusivas por medio de los nodos *network*. Esta capacidad de fusión estructural demostró ser una herramienta eficaz para identificar regiones de interés común, evaluar la estabilidad de la búsqueda frente a condiciones experimentales contrastantes, y detectar divergencias inducidas por decisiones de política de sintonización como el uso del *soft-restart*.

En conjunto, los resultados experimentales validan la utilidad del modelo STN-i como herramienta de análisis estructural y visual del proceso de configuración automática mediante *irace*. Las métricas propuestas, junto con la flexibilidad del modelo y la capacidad de generar representaciones visuales informativas, permiten estudiar propiedades profundas del proceso de búsqueda, superando las limitaciones de los enfoques centrados exclusivamente en la calidad final de las configuraciones encontradas. Así, STN-i se posiciona como una solución complementaria de alto valor para la comprensión, comparación y diseño de procesos de sintonización eficientes y explicables.

Sin embargo, es importante destacar que, si bien el modelo STN-i demuestra ser una herramienta expresiva y eficaz para el análisis de *irace* en distintos contextos, su uso efectivo requiere un conocimiento profundo del dominio del problema, del algoritmo sintonizado y del propio proceso de sintonización. La interpretación de las redes generadas depende directamente de múltiples decisiones metodológicas adoptadas durante su construcción, tales como el esquema de discretización aplicado, la función de evaluación seleccionada, la estrategia de etiquetado de nodos y el algoritmo de disposición gráfica utilizado. Estas decisiones no son neutrales: condicionan la estructura resultante y afectan directamente la lectura e interpretación de las trayectorias.

En consecuencia, el valor analítico de STN-i no radica únicamente en su formalismo estructural, sino también en la articulación informada entre el conocimiento técnico del usuario y la semántica visual de las redes. Solo bajo una mirada crítica y contextualizada es posible aprovechar plenamente el potencial explicativo del modelo, evitando interpretaciones erróneas o simplificaciones indebidas. Este aspecto refuerza el carácter complementario de STN-i dentro de un marco más amplio de análisis de procesos de configuración automática, en el cual el criterio experto sigue siendo esencial para guiar, interpretar y validar las observaciones obtenidas.

5.1. Trabajo futuro

Si bien el modelo STN-i ha demostrado ser una herramienta versátil y potente para el análisis estructural del proceso de sintonización automática mediante *irace*, existen diversas líneas de trabajo futuro que permitirían profundizar en sus capacidades analíticas y extender su aplicabilidad a nuevos escenarios. Una de las extensiones más prometedoras corresponde a la incorporación de una dimensión temporal explícita en la representación de las redes. Aunque actualmente se cuenta con información relativa a la calidad de las configuraciones a través del posicionamiento topológico de los nodos, el modelo no incorpora una noción directa del avance temporal del proceso de sintonización. Considerando que la herramienta *irace-STN-i* permite recuperar información detallada de las iteraciones en las que cada configuración fue evaluada, sería factible generar visualizaciones segmentadas por iteración o por porcentaje de avance, lo cual abriría la posibilidad de construir representaciones dinámicas o *imágenes* parciales del proceso de exploración. Esta propuesta también podría extenderse a la construcción progresiva de redes merged STN-i, permitiendo evaluar cuándo y cómo emergen las regiones compartidas o divergentes entre ejecuciones contrastantes.

En paralelo, otra línea de desarrollo consiste en enriquecer el conjunto de métricas disponibles con indicadores que respondan a propiedades específicas de la estructura, tanto a nivel local como global. Si bien las métricas actuales permiten caracterizar aspectos fundamentales como la proporción de nodos elite, la longitud de trayectorias o la conectividad hacia los mejores nodos, aún existen propiedades relevantes no capturadas de forma explícita. En particular, sería interesante incorporar métricas asociadas a la intensidad o fuerza de ciertas trayectorias —por ejemplo, considerando la frecuencia con que se repite una transición entre nodos a lo largo de múltiples ejecuciones— así como indicadores de interconexión entre grupos específicos de nodos, como enlaces frecuentes entre elites, entre nodos compartidos y exclusivos, o entre nodos inicialmente considerados de bajo rendimiento pero que se integran estructuralmente a zonas clave de la red. Asimismo, resulta valioso explorar métricas de tendencia que midan la transformación estructural de ciertas configuraciones, como el cambio de rol topológico a lo largo del tiempo, o la transición de nodos originalmente etiquetados como START a posiciones END en la red, como reflejo de persistencia o supervivencia exploratoria.

Además, una mejora relevante sería extender el marco actual de análisis para permitir la evaluación de métricas sobre redes filtradas o visualizaciones parciales, tales como aquellas que excluyen los nodos regulares de inicio. Esta capacidad permitiría cuantificar propiedades estructurales en subconjuntos más significativos de la red, eliminando el ruido asociado a configuraciones descartadas tempranamente. Este enfoque también facilitaría la exploración comparativa entre vistas, contribuyendo a un análisis más detallado y contextualizado de regiones relevantes dentro de la red.

Más allá de las mejoras al modelo, resulta igualmente valioso continuar explorando nuevos análisis estructurales aplicados a dominios complejos. Una dirección concreta consiste en replicar el tipo de estudio realizado en este trabajo para el caso de PSO-X, ampliando el

conjunto de instancias utilizadas y evaluando el comportamiento del modelo bajo variantes adicionales del proceso de sintonización, tales como la modificación de componentes internos de *irace* (por ejemplo, reemplazando el test de Friedman por t-student). Estos nuevos experimentos permitirían estudiar con mayor profundidad el impacto de los mecanismos estadísticos, las decisiones de muestreo y las condiciones de evaluación sobre la estructura de las trayectorias generadas, enriqueciendo el entendimiento del vínculo entre los componentes internos del configurador y las dinámicas resultantes.

Junto con las mejoras propuestas al modelo STN-i, también se identifican oportunidades concretas de extensión sobre el propio sintonizador *irace*, orientadas a fortalecer su capacidad de análisis estructural y trazabilidad. En primer lugar, se propone ampliar el contenido del archivo de salida generado por *irace* (*.Rdata*), incorporando información detallada sobre las configuraciones activas en cada iteración del proceso. Esta información incluiría el listado completo de identificadores de configuraciones presentes en cada iteración, junto con el número de evaluaciones completadas por cada una antes de ser eventualmente descartada. Esta trazabilidad permitiría caracterizar la supervivencia relativa de las configuraciones dentro de cada iteración y facilitaría la extracción de patrones sin necesidad de recurrir a niveles avanzados de depuración.

En segundo lugar, dado lo observado en las pruebas realizadas con los algoritmos objetivo, donde se identificaron patrones estructurales complejos en dominios con mayor número de parámetros y relaciones condicionales (como PSO-X y ACOTSP), se plantea una propuesta más ambiciosa orientada a mejorar el funcionamiento interno de *irace*. En particular, se propone incorporar de forma opcional un mecanismo para detectar y aprovechar correlaciones entre parámetros durante el proceso de sintonización, con foco en la etapa de selección estadística de configuraciones élite. Esta funcionalidad permitiría complementar los criterios tradicionales de descarte —como el test de Friedman— con información estructural obtenida a partir del historial de configuraciones evaluadas. Para ello, se sugiere implementar un módulo de análisis que calcule dependencias empíricas entre parámetros, empleando métricas como la correlación de *Pearson*, la correlación por rangos de *Spearman*, la *mutual information* (información mutua) o análisis de co-ocurrencias basados en la prueba de *chi-cuadrado*. Esta información permitiría evitar la eliminación de configuraciones que, si bien pueden no destacar de forma individual, representan combinaciones de parámetros sinérgicas o con interacciones relevantes.

En la práctica, este módulo funcionaría como una capa adicional al proceso de selección: luego de aplicar el test estadístico tradicional, se podría revisar qué configuraciones aportan valor estructural al espacio de búsqueda, ya sea ajustando el umbral de descarte o preservando ciertos candidatos. Para garantizar su utilidad y eficiencia, se propone activar este mecanismo posterior a un número mínimo de iteraciones, una vez acumulada suficiente información para detectar patrones significativos; no obstante, también podría utilizarse desde el inicio mediante estimaciones preliminares. Si bien esta integración conllevaría un costo computacional adicional, su carácter opcional y parametrizable permitiría mantener la flexibilidad del framework. Además, esta línea de desarrollo podría abrir la puerta a futuras adaptacio-

nes del proceso de generación de configuraciones, haciéndolo más sensible a la estructura del espacio paramétrico explorado.

REFERENCIAS BIBLIOGRÁFICAS

- [Aarts y Korst, 1989] Aarts, E. y Korst, J. (1989). *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley & Sons, Inc.
- [Aarts y Lenstra, 1997] Aarts, E. y Lenstra, J. K. (1997). *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., USA, 1st edición.
- [Abdel-Basset et al., 2018] Abdel-Basset, M., Abdel-Fatah, L., y Sangaiah, A. K. (2018). Meta-heuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185–231.
- [Ansótegui et al., 2015] Ansótegui, Carlos and Malitsky, Yuri and Samulowitz, Horst and Sellmann, Meinolf and Tierney, Kevin and others (2015). Model-based genetic algorithms for algorithm configuration. En *IJCAI*, pp. 733–739.
- [Balaprakash et al., 2007] Balaprakash, P., Birattari, M., y Stützle, T. (2007). Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. En *Hybrid Metaheuristics: 4th International Workshop, HM 2007, Dortmund, Germany, October 8-9, 2007. Proceedings 4*, pp. 108–122. Springer.
- [Ballardini, 2018] Ballardini, A. L. (2018). A tutorial on particle swarm optimization clustering. Imagen recuperada de ResearchGate.
- [Birattari et al., 2010] Birattari, M., Yuan, Z., Balaprakash, P., y Stützle, T. (2010). F-race and iterated f-race: An overview. *Experimental methods for the analysis of optimization algorithms*, pp. 311–336.
- [Blackwell y Branke, 2004] Blackwell, T. y Branke, J. (2004). Multi-swarm optimization in dynamic environments. En Raidl, G. R., Cagnoni, S., Branke, J., Corne, D. W., Drechsler, R., Jin, Y., Johnson, C. G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G. D., y Squillero, G., editores, *Applications of Evolutionary Computing*, pp. 489–500, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Blum y Roli, 2003] Blum, C. y Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308.
- [Bullnheimer et al., 1999] Bullnheimer, B., Hartl, R. F., y Strauss, C. (1999). A new rank based version of the ant system—a computational study. *Central European Journal of Operations Research*, 7(1).
- [Cáceres et al., 2017] Cáceres, L. P., López-Ibáñez, M., Hoos, H., y Stützle, T. (2017). An experimental study of adaptive capping in irace. En *International Conference on Learning and Intelligent Optimization*, pp. 235–250. Springer.

- [Camacho-Villalón et al., 2022] Camacho-Villalón, C. L., Dorigo, M., y Stützle, T. (2022). Pso-x: A component-based framework for the automatic design of particle swarm optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 26(3):402–416.
- [Cantu-Paz, 2000] Cantu-Paz, E. (2000). *Efficient and accurate parallel genetic algorithms*, volumen 1. Springer Science & Business Media.
- [Cleghorn y Ochoa, 2021] Cleghorn, C. W. y Ochoa, G. (2021). Understanding parameter spaces using local optima networks: a case study on particle swarm optimization. En *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '21, p. 1657–1664, New York, NY, USA. Association for Computing Machinery.
- [Clerc y Kennedy, 2002] Clerc, M. y Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73.
- [Crama et al., 1995] Crama, Y., Kolen, A. W. J., y Pesch, E. J. (1995). *Local search in combinatorial optimization*, pp. 157–174. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Daley, 2015] Daley, J. (2015). We're terrible at distinguishing real and fake schools of fish. *Smithsonian Magazine*, image by: Reinhard Dirscherl/WaterFrame/Corbis.
- [Doerr, 2019] Doerr, C. (2019). Dynamic parameter choices in evolutionary computation. En *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 890–922.
- [Dorigo y Gambardella, 1997] Dorigo, M. y Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.
- [Dorigo et al., 1996] Dorigo, M., Maniezzo, V., y Coloni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41.
- [Edmonds, 1971] Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical programming*, 1:127–136.
- [Fruchterman y Reingold, 1991] Fruchterman, T. M. y Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.
- [Garey y Johnson, 1979] Garey, M. R. y Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.
- [Geman y Geman, 1984] Geman, S. y Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

- [Glover, 1986] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers Operations Research*, 13(5):533–549. Applications of Integer Programming.
- [Glover, 1990] Glover, F. (1990). Tabu search—part ii. *ORSA Journal on computing*, 2(1):4–32.
- [Gomes Pereira de Lacerda et al., 2021] Gomes Pereira de Lacerda, M., de Araujo Pessoa, L. F., Buarque de Lima Neto, F., Ludermir, T. B., y Kuchen, H. (2021). A systematic literature review on general parameter control for evolutionary and swarm-based algorithms. *Swarm and Evolutionary Computation*, 60:100777.
- [Griva et al., 2008] Griva, I., Nash, S. G., y Sofer, A. (2008). *Linear and nonlinear optimization 2nd edition*. SIAM.
- [Herrera et al., 2010] Herrera, Francisco and Lozano, Manuel and Molina, D and others (2010). Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. *Last accessed: July*.
- [Hinterding et al., 1999] Hinterding, R., Michalewicz, Z., y Eiben, A. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141.
- [Holland, 1992] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
- [Hu et al., 2023] Hu, T., Ochoa, G., y Banzhaf, W. (2023). Phenotype search trajectory networks for linear genetic programming. En Pappa, G., Giacobini, M., y Vasicek, Z., editores, *Genetic Programming*, pp. 52–67, Cham. Springer Nature Switzerland.
- [Hutter, 2009] Hutter, F. (2009). *Automated configuration of algorithms for solving hard computational problems*. Tesis doctoral, University of British Columbia.
- [Hutter et al., 2011] Hutter, F., Hoos, H. H., y Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. En *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pp. 507–523. Springer.
- [Hutter et al., 2009] Hutter, F., Hoos, H. H., Leyton-Brown, K., y Stützle, T. (2009). Paramils: an automatic algorithm configuration framework. *Journal of artificial intelligence research*, 36:267–306.
- [Karafotias et al., 2015] Karafotias, G., Hoogendoorn, M., y Eiben, A. E. (2015). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2):167–187.

- [Kauffman, 1989] Kauffman, S. A. (1989). Adaptation on rugged fitness landscapes. *Studies in the sciences of complexity*, 1:527–618.
- [Kennedy y Eberhart, 1995] Kennedy, J. y Eberhart, R. (1995). Particle swarm optimization. En *Proceedings of ICNN'95 - International Conference on Neural Networks*, volumen 4, pp. 1942–1948 vol.4.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., y Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- [Koopmans y Beckmann, 1957] Koopmans, T. C. y Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pp. 53–76.
- [Lavinas et al., 2022] Lavinas, Y., Aranha, C., y Ochoa, G. (2022). Search trajectories networks of multiobjective evolutionary algorithms. En Jiménez Laredo, J. L., Hidalgo, J. I., y Babaagba, K. O., editores, *Applications of Evolutionary Computation*, pp. 223–238, Cham. Springer International Publishing.
- [Lawler, 1985] Lawler, E. L. (1985). The traveling salesman problem: a guided tour of combinatorial optimization. *Wiley-Interscience Series in Discrete Mathematics*.
- [Liang et al., 2013] Liang, Jing J and Qu, Bo Y and Suganthan, Ponnuthurai N and others (2013). Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635(2):2014.
- [López-Ibáñez et al., 2016] López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., y Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- [Montero et al., 2014] Montero, E., Riff, M.-C., y Neveu, B. (2014). A beginner's guide to tuning methods. *Applied Soft Computing*, 17:39–51.
- [Narvaez-Teran et al., 2021] Narvaez-Teran, V., Ochoa, G., y Rodríguez-Tello, E. (2021). Search trajectory networks applied to the cyclic bandwidth sum problem. *IEEE Access*, 9:151266–151277.
- [Ochoa, 2021] Ochoa, G. (2021). Stns: Search trajectory networks (repositorio de código y figuras). Repositorio en GitHub. Último acceso: 22-05-2025.
- [Ochoa et al., 2021] Ochoa, G., Malan, K. M., y Blum, C. (2021). Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics. *Applied Soft Computing*, 109:107492.
- [Ochoa et al., 2014] Ochoa, G., Verel, S., Daolio, F., y Tomassini, M. (2014). *Local Optima Networks: A New Model of Combinatorial Fitness Landscapes*, pp. 233–262. Springer Berlin Heidelberg, Berlin, Heidelberg.

- [Ong et al., 2006] Ong, Y.-S., Lim, M.-H., Zhu, N., y Wong, K.-W. (2006). Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1):141–152.
- [Papadimitriou, 1991] Papadimitriou, C. H. (1991). On selecting a satisfying truth assignment. En [1991] *Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pp. 163–169. IEEE Computer Society.
- [Pearl, 1984] Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc.
- [Pei et al., 2012] Pei, Y., Wang, W., y Zhang, S. (2012). Basic ant colony optimization. En 2012 *International Conference on Computer Science and Electronics Engineering*, volumen 1, pp. 665–667.
- [Pérez Cáceres et al., 2014] Pérez Cáceres, L., López-Ibáñez, M., y Stützle, T. (2014). An analysis of parameters of irace. En Blum, C. y Ochoa, G., editores, *Evolutionary Computation in Combinatorial Optimisation*, pp. 37–48, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Riff y Montero, 2013] Riff, M.-C. y Montero, E. (2013). A new algorithm for reducing meta-heuristic design effort. En 2013 *IEEE Congress on Evolutionary Computation*, pp. 3283–3290. IEEE.
- [Riveros et al., 2024] Riveros, M., Rojas-Morales, N., Montero, E., y Ochoa, G. (2024). Understanding search trajectories in parameter tuning. En *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 778–786.
- [Rosenkrantz et al., 1977] Rosenkrantz, D. J., Stearns, R. E., y Lewis, II, P. M. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581.
- [Russell y Norvig, 2020] Russell, S. y Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, 4 edición.
- [Saifullah Hussin y Stützle, 2014] Saifullah Hussin, M. y Stützle, T. (2014). Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances. *Computers Operations Research*, 43:286–291.
- [Sarti et al., 2022] Sarti, S., Adair, J., y Ochoa, G. (2022). Neuroevolution trajectory networks of the behaviour space. En Jiménez Laredo, J. L., Hidalgo, J. I., y Babaagba, K. O., editores, *Applications of Evolutionary Computation*, pp. 685–703, Cham. Springer International Publishing.
- [Schrijver, 2003] Schrijver, A. (2003). *Combinatorial Optimization: Polyhedra and Efficiency*, volumen 24 de *Algorithms and Combinatorics*. Springer.

- [Shi y Eberhart, 1998] Shi, Y. y Eberhart, R. (1998). A modified particle swarm optimizer. En *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pp. 69–73.
- [Simon y Newell, 1971] Simon, H. A. y Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American psychologist*, 26(2):145.
- [Stützle, 2002] Stützle, T. (2002). Acotsp: A software package of various ant colony optimization algorithms applied to the symmetric traveling salesman problem. URL <http://www.aco-metaheuristic.org/aco-code>.
- [Stützle y Hoos, 1998] Stützle, T. y Hoos, H. (1998). Improvements on the ant-system: Introducing the max-min ant system. En *Artificial Neural Nets and Genetic Algorithms*, pp. 245–249, Vienna. Springer Vienna.
- [Suganthan et al., 2005] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., y Tiwari, S. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report*, 2005005(2005):2005.
- [Treimun-Costa et al., 2020] Treimun-Costa, G., Montero, E., Ochoa, G., y Rojas-Morales, N. (2020). Modelling parameter configuration spaces with local optima networks. En *Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO '20*, p. 751–759, New York, NY, USA. Association for Computing Machinery.
- [White et al., 2003] White, T., Kaegi, S., y Oda, T. (2003). Revisiting elitism in ant colony optimization. En Cantú-Paz, E., Foster, J. A., Deb, K., Davis, L. D., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Standish, R., Kendall, G., Wilson, S., Harman, M., Wegener, J., Dasgupta, D., Potter, M. A., Schultz, A. C., Dowsland, K. A., Jonoska, N., y Miller, J., editores, *Genetic and Evolutionary Computation — GECCO 2003*, pp. 122–133, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Zambrano-Bigiarini et al., 2013] Zambrano-Bigiarini, M., Clerc, M., y Rojas, R. (2013). Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. En *2013 IEEE Congress on Evolutionary Computation*, pp. 2337–2344.