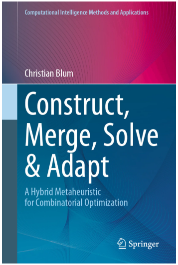
CMSA: A Hybrid Metaheuristic for Combinatorial Optimization

Christian Blum

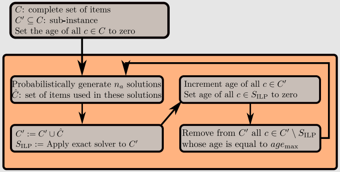
Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona



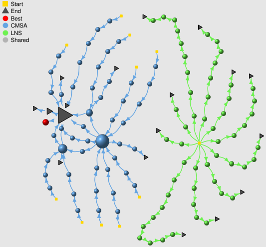
CSIC
Consejo Superior de Investigaciones Científicas

Presentation Content

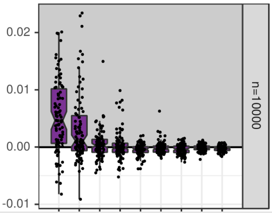
CMSA Algorithm



Search Trajectory Networks



Appl. to optimization problem



Outlook

Personal introduction

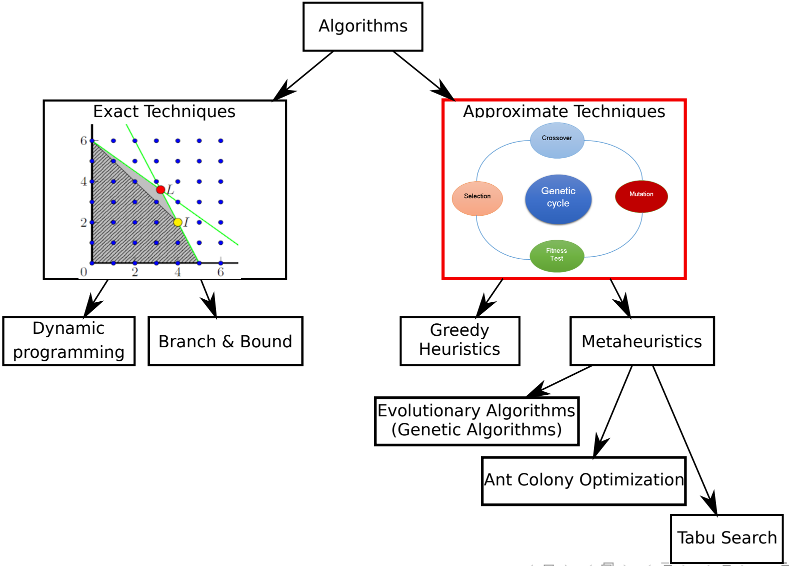
CSIC: Spanish National Research Council

- Largest public institution dedicated to research in Spain (created in 1939)
- Third-largest in Europe
- 6% of all research staff in Spain work for the CSIC
- 20% of the scientific production in Spain is from the CSIC

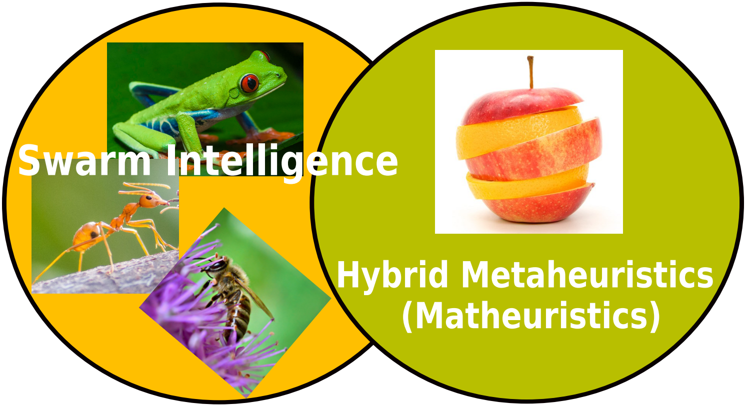
Artificial Intelligence Research Institute (IIIA)

- 35 tenured scientists (of three different ranks)
- Around 65 additional staff members (post-docs, Ph.D. students, technicians, administration)
- Research lines: machine learning, optimization, logic and reasoning, multi-agent systems

Combinatorial optimization: algorithms



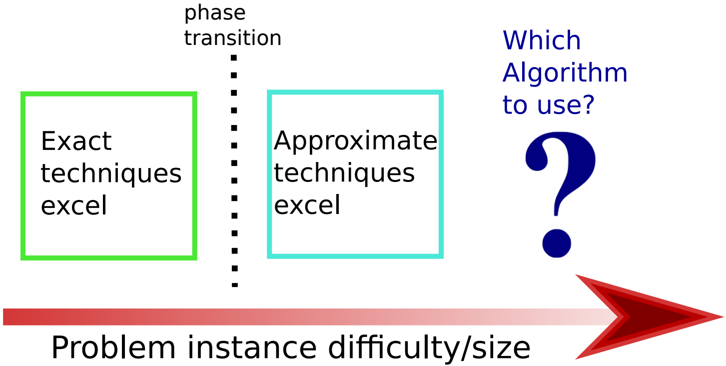
Our research topics: algorithm-oriented



Topic of today: CMSA

A prominent example of our recent work on hybrid metaheuristics based on problem instance reduction

Hybrid metaheuristics: reasons for being



Note

Hybrid algorithms that exploit synergies between exact and approximate algorithms often excel in the context of large-scale problems.

Problem instance reduction

Observation

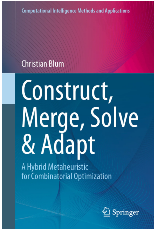
- An exact technique (such as a black-blox MIP solver) may not be directly applicable for solving large-scale problem instances
- **Nevertheless:** It might still be useful for the application to reduced instances of the original problem instances

Idea is not new

- Large neighborhood search (LNS)
  1. Based on partial solution destruction
  2. Local branching, corridor method
- Decomposition approaches such as POPMUSIC
- Generate-and-solve (GS) framework
- Set covering based approaches: for example in the context of VRPs
- Solution merging (resp. optimal recombination) in EAs

Contruct, Merge, Solve & Adapt, CMSA

Initial CMSA team



CMSA Book 2024

Winners of the **SEIO-FBBVA award 2021** for the ***best methodological contribution to Operations Research***.

- Spanish Society of Statistics and Operations Research
- Fundación BBVA BBVA Foundation (Banco Bilbao Vizcaya Argentaria)

CMSA: main idea

Observation

In the presence of a large number of solution components, many of them only form part of low-quality solutions

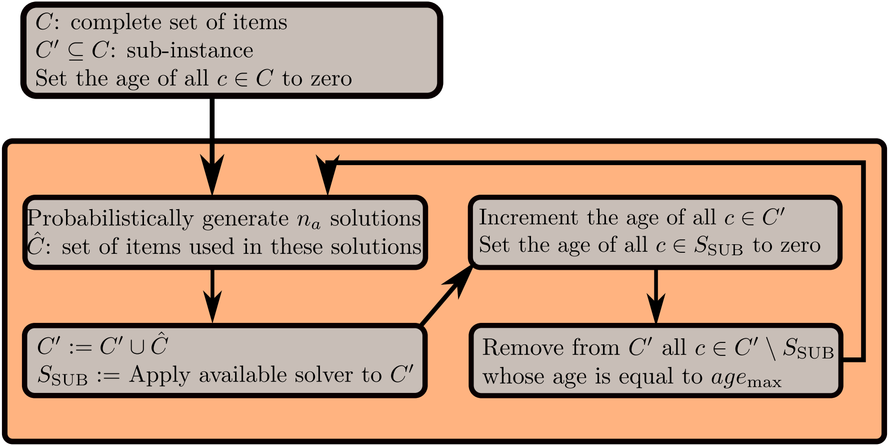
Idea

Exclude these presumably bad solution components before applying your solver

Steps of one CMSA iteration

- Iteratively generate presumably good solutions in a probabilistic way
- Assemble a sub-instance from the used solution components
- Solve the sub-instance by means of an available solver
- Delete useless solution components from the sub-instance

CMSA: flow diagram



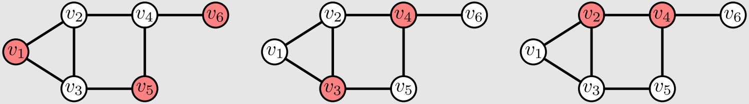
Topic 1: CMSA Application to Minimum Dominating Set (MDS)

Example Application: Min. Dom. Set

Definition: Minimum Dominating Set (MDS) problem

- **Given:** An undirected graph $G = (V, E)$.
- **Search space:** Every set $S \subseteq V$ such that every $v \in V \setminus S$ has at least one neighbor in S.
- **Objective function:** $F(S) := |S|$ (cardinality of set S)
- **Optimization objective:** minimization

Examples of solutions



Example Application: Min. Dom. Set

ILP model of the MDS Problem

$$
\begin{aligned}
\min \quad & \sum_{v \in V} x_v \\
\text{subject to} \quad & x_v + \sum_{u \in N(v)} x_u \geq 1 && \forall\, v \in V \\
& x_v \in \{0, 1\} && \forall\, v \in V
\end{aligned}
$$

What do we need for implementing a CMSA?

1. Define the **set of solution components**
2. A **greedy heuristic** used in a probabilistic way for generating solutions
3. An **approach for solving the sub-instance** of each iteration:
   - An exact technique
   - A metaheuristic approach

CMSA application to the MDS

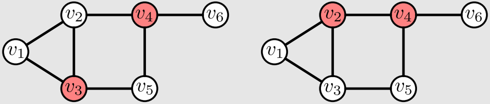
Definition of solution components

- **Variant 1:** for each vertex v_i we introduce a solution component c_i
- **Variant 2:** each combination of a decision variable x_i with one of its values is a solution component. That is, for all $v_i \in V$:
  - x_i and value 0 $\rightarrow$ solution component $c_{i,0}$
  - x_i and value 1 $\rightarrow$ solution component $c_{i,1}$

MDS Greedy Heuristic

- Add one vertex at each construction step
- Among the unselected vertices: choose the one that covers the largest number of the so-far uncovered vertices
- Apply the heuristic in a probabilistic way

Example: two solutions constructed per iteration



CMSA Variant 1: solving the sub-instance

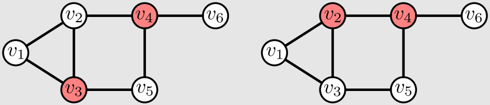
Solution 1:	c_1	c_2	c_3	c_4	c_5	c_6
Solution 2:	c_1	c_2	c_3	c_4	c_5	c_6

Therefore: add the following constraints to the ILP model

$$x_i = 0 \quad \text{for } v_1, v_5, v_6$$

CMSA Variant 2: Sub-instance Solving

Example: two solutions constructed per iteration



CMSA Variant 2: solving the sub-instance

Solution 1:	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$	$c_{4,0}$	$c_{5,0}$	$c_{6,0}$
	$c_{1,1}$	$c_{2,1}$	$c_{3,1}$	$c_{4,1}$	$c_{5,1}$	$c_{6,1}$
Solution 2:	$c_{1,0}$	$c_{2,0}$	$c_{3,0}$	$c_{4,0}$	$c_{5,0}$	$c_{6,0}$
	$c_{1,1}$	$c_{2,1}$	$c_{3,1}$	$c_{4,1}$	$c_{5,1}$	$c_{6,1}$

Therefore: add the following constraints to the ILP model

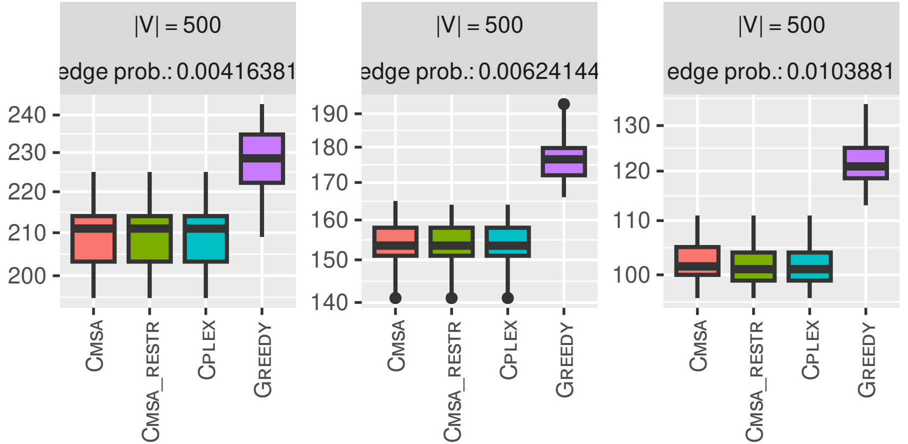
$x_i = 0 \quad \text{for } v_1, v_5, v_6$ $\qquad$ $x_i = 1 \quad \text{for } v_4$

MDS Problem: Experimental Setup

- **Problem instances:** Erdős-Rényi random graphs with {500, 1000, 1500, 2000} nodes and three different densities
- **Applied algorithms:** CMSA, CMSA_RESTR, CPLEX, GREEDY
- **Algorithm tuning:** with `irace`, a scientific tool for parameter tuning
- **Parameters considered for tuning:**
  1. CMSA algorithms: number of solution constructions per iteration, degree of randomness, maximum age for solution components
  2. CPLEX within CMSA: time limit, whether to use warm start, whether to use heuristic emphasis, whether to abort a run when best-so-far solution improved

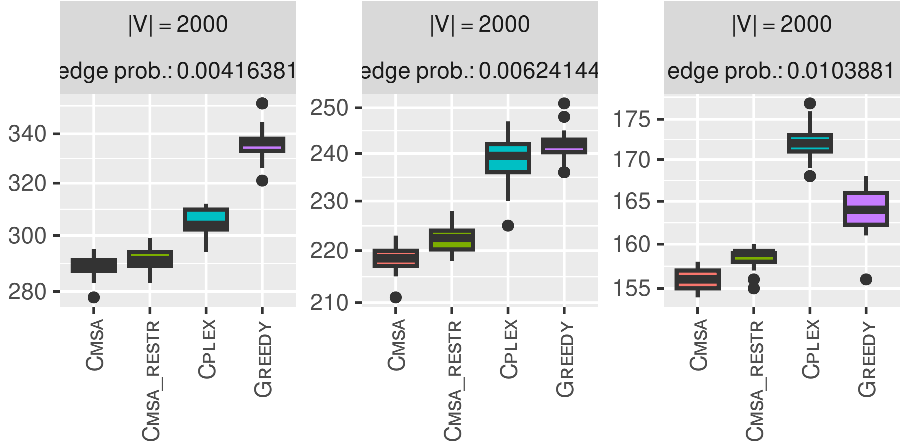
MDS Results

Erdős-Rényi graphs, 500 nodes



MDS Results

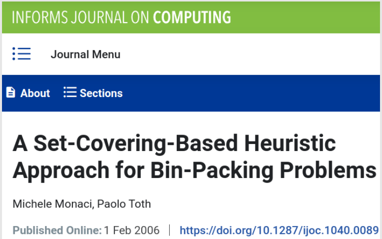
Erdős-Rényi graphs, 2000 nodes



Topic 2: CMSA for Set-Covering Based Problem Transformations

Solution Components: CMSA Variant 3

Expressing a problem as a **set covering problem**


INFORMS JOURNAL ON **COMPUTING**

Journal Menu

About Sections

A Set-Covering-Based Heuristic Approach for Bin-Packing Problems

Michele Monaci, Paolo Toth

Published Online: 1 Feb 2006 | https://doi.org/10.1287/ijoc.1040.0089


Home > Journal of Combinatorial Optimization >

Article

A set covering approach for multi-depot train driver scheduling

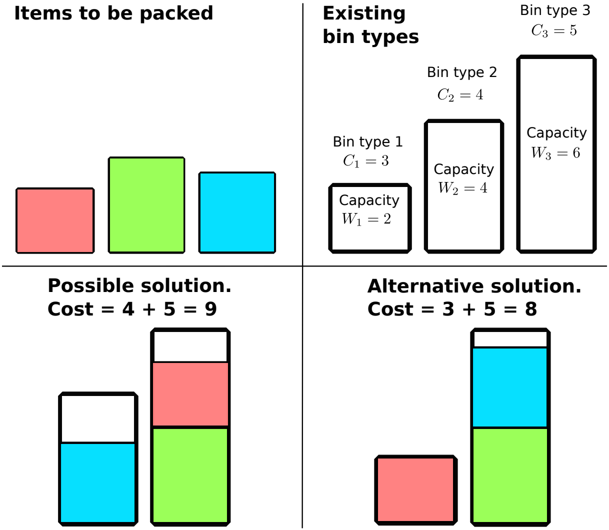
Published: 07 April 2013

Volume 29, pages 636–654, (2015)

Opportunity for CMSA

Define solution components to be complete sets of the set covering model of a problem.

Variable Sized Bin Packing (VSBP)



Variable Sized Bin Packing (VSBP)

Standard ILP Model

$$\min \sum_{j=1}^{n} \sum_{k=1}^{m} c_k \cdot y_{jk}$$

$$
\begin{aligned}
\textbf{s.t.:} \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, \ldots, n \\
& \sum_{k=1}^{m} y_{jk} \leq 1 \quad \text{for } j = 1, \ldots, n \\
& \sum_{i=1}^{n} w_i \cdot x_{ij} \leq \sum_{k=1}^{m} W_k \cdot y_{jk} \quad \text{for } j = 1, \ldots, n \\
& x_{ij} \in \{0, 1\} \quad \text{for } i, j = 1, \ldots, n \\
& y_{jk} \in \{0, 1\} \quad \text{for } j = 1, \ldots, n \\
& \qquad\qquad\qquad\qquad k = 1, \ldots, m
\end{aligned}
$$

Variables of the model

- x_{ij} is set to 1 iff item i is assigned to bin j
- y_{jk} is set to 1 iff bin j is assigned bin type k

VSBP: Set Covering ILP Model

Definitions for the Set Covering ILP Model

- Let $\mathcal{B}$ be the set of all possible bins
- The weight w_b of a bin $b \in \mathcal{B}$ is the sum of the weights of its items
- Assign to $b \in \mathcal{B}$ a cost c_b which corresponds to the lowest-cost bin type possible for b
- Let $\mathcal{B}_i \subset \mathcal{B}$ be the set of bins that contain item i

Set Covering ILP Model

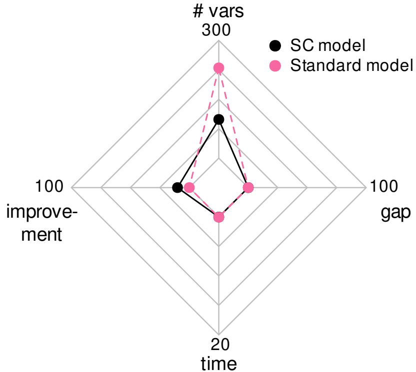
$$\textbf{min} \sum_{b \in \mathcal{B}} c_b \cdot x_b$$

$$\textbf{s.t.:} \quad \sum_{b \in \mathcal{B}_i} x_b \geq 1 \quad \text{for } i = 1, \ldots, n$$

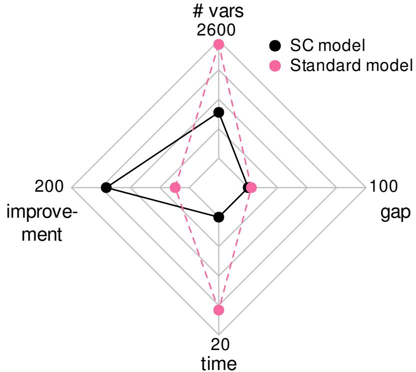
$$x_b \in \{0, 1\} \quad \text{for all } b \in \mathcal{B}$$

VSBPP: Set Covering ILP Model

Problem instance with **100 items**



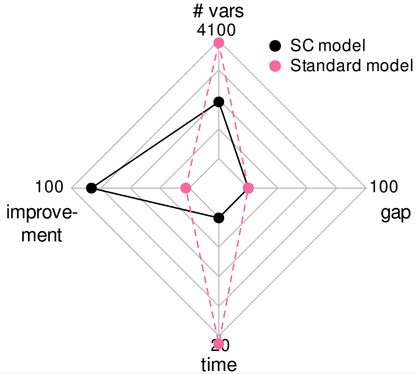
Sub-instance based on 2 solutions



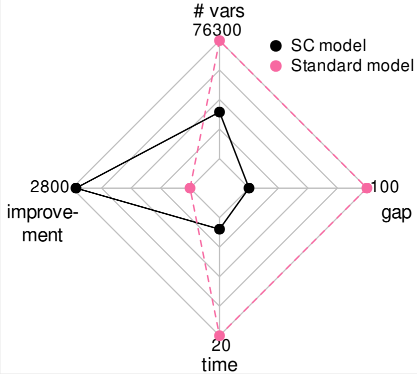
Sub-instance based on 50 solutions

VSBPP: Set Covering ILP Model

Problem instance with **2000 items**



Sub-instance based on 2 solutions



Sub-instance based on 50 solutions

VSBPP: Implementation and Setup

VSBPP State of the art

Computers & Operations Research
Volume 39, Issue 5, May 2012, Pages 1097-1108

Variable neighbourhood search for the variable sized bin packing problem ☆

Vera Hemmelmayr [a], Verena Schmid [a], Christian Blum [b]

Computers & Operations Research
Volume 36, Issue 10, October 2009, Pages 2877-2884

Heuristics for the variable sized bin-packing problem

Mohamed Haouari [a b], Mehdi Serairi [b]

Implementation Characteristics

- **Heuristic:** based on ordering the items and placing them in this order into bins according to least-increasing cost
- **Sub-instance solving:** solve the set covering model, followed by a simple repair procedure

Problem instances

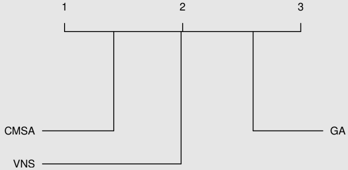
- 150 instances
- Between 100 and 2000 items
- 7 bin types

VSBPP: Comparison / CD Plots

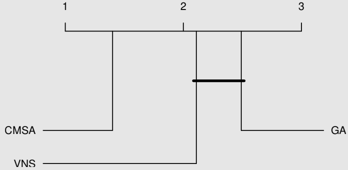
Overall result

68 new best-known solutions. Only worse for 7 instances.

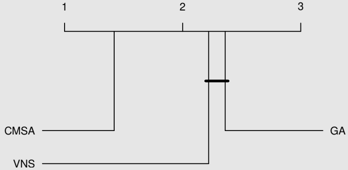
All instances



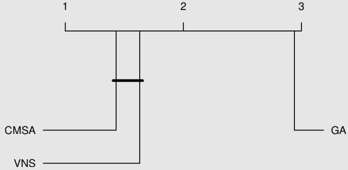
Only B1 instances



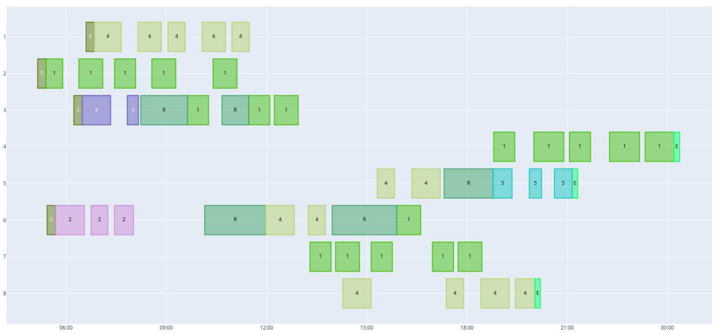
Only B2 instances



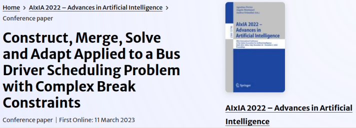
Only B3 instances



Real World Problem from Austria



Publication


Home > AIxIA 2022 – Advances in Artificial Intelligence >

Conference paper

Construct, Merge, Solve and Adapt Applied to a Bus Driver Scheduling Problem with Complex Break Constraints

Conference paper | First Online: 11 March 2023

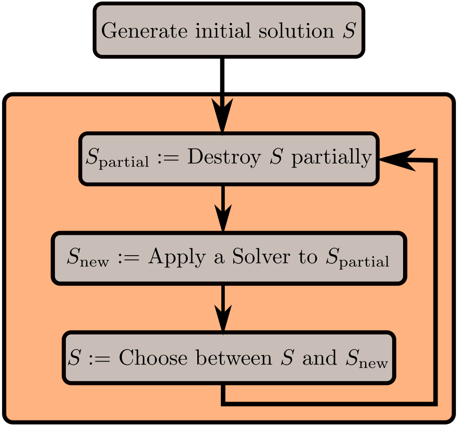
AIxIA 2022 – Advances in Artificial Intelligence

BSD: Sample Results

Size	CMSA	SA	HC	TS	CH-PR	GIHH	L-GIHH	B&P *
10	14879,7	**14739,6**	14988,4	15036,4	14956,2	14847,4	14810,6	14709,2
20	**30745,9**	30971,0	31275,6	31248,4	30896,7	30892,2	30810,8	30294,8
30	**50817,2**	51258,0	51917,4	51483,0	51331,4	51059,4	51037,6	49846,4
40	**68499,9**	69379,8	71337,6	69941,2	69182,9	68988,4	69022,2	67000,4
50	**86389,2**	87557,4	87262,4	87850,6	87394,3	87184,4	87145,2	84341,0
60	**102822,9**	104333,0	104296,4	104926,2	103921,5	103491,6	103467,3	99727,0
70	**121141,9**	123225,6	123304,0	123632,2	122502,9	122198,6	122321,8	118524,2
80	**138760,3**	140914,0	140508,0	140482,4	139931,8	139648,2	139551,9	134513,8
90	**155078,3**	157426,0	156862,4	156296,4	155520,8	155560,8	155649,6	150370,8
100	**171786,7**	174501,8	172909,0	172916,0	171901,0	171879,8	172763,7	172582,2
150	**263387,7**	266705,5	265492,3	265654,8	-	-	-	-
200	**349017,0**	354408,4	353494,9	350747,2	-	-	-	-
250	**439234,5**	446525,0	446000,9	443845,8	-	-	-	-

Topic 3: Differences between CMSA and LNS (Large Neighborhood Search)

CMSA: Difference to LNS



LNS: recent surey

Handbook of Metaheuristics pp 99–127 | Cite as

Home > Handbook of Metaheuristics > Chapter

Large Neighborhood Search

David Pisinger & Stefan Ropke

Comparison CMSA vs. LNS

ELSEVIER

European Journal of Operational Research

Volume 290, Issue 1, 1 April 2021, Pages 36-56

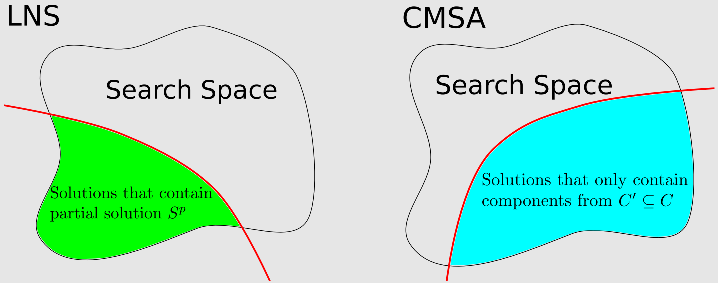
Discrete Optimization

A comparative analysis of two matheuristics by means of merged local optima networks

Christian Blum [a], Gabriela Ochoa [b]

CMSA: Difference to LNS

How is the original problem instance reduced?



- **LNS:** Partial destruction of the incumbent solution
- **CMSA:** Probabilistically adding solution components
- **Our intuition:** for subset selection problems, CMSA better than LNS when solutions contain few components

Comarison CMSA vs. LNS

CMSA/LNS for the Multi-dimensional knapsack problem (MDKP)

What do we need?

1. A constructive heuristic for generating solutions
2. A way of solving sub-instances (in this case: ILP model)

ILP model for the MDKP

$$\textbf{max} \quad \sum_{i \in C} p_i \cdot x_i$$

$$\textbf{subject to:} \quad \sum_{c_i \in C} r_{i,k} \cdot x_i \leq \text{cap}_k \qquad \forall k \in K$$

$$x_i \in \{0, 1\} \qquad \forall c_i \in C$$

Application of CPLEX within LNS and CMSA

- LNS: fix $x_i = 1 \; \forall \; c_i \in S_{\text{partial}}$.
- CMSA: replace C with C'

MDKP instances: *tightness*

Important parameter for instance generation: *tightness* $(0 \leq \alpha \leq 1)$

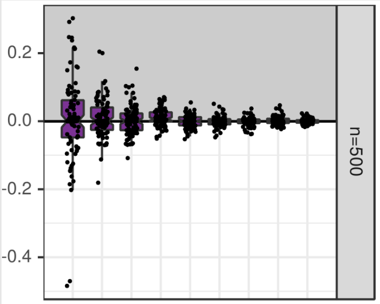
- When α is close to zero: capacities are low and valid solution only contain very few items
- When α is close to one: capacities are very high and solutions contain nearly all item

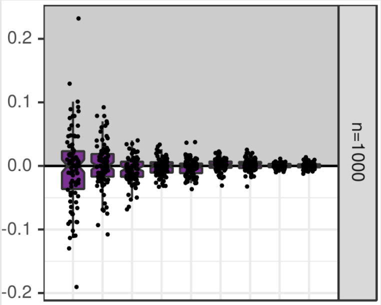
The plan

- Apply both LNS and CMSA to instances from the whole tightness range
- Both algorithms are tuned with `irace` seperately for instances of each considered tightness

MDKP Problem: CMSA vs. LNS

Rather small instances



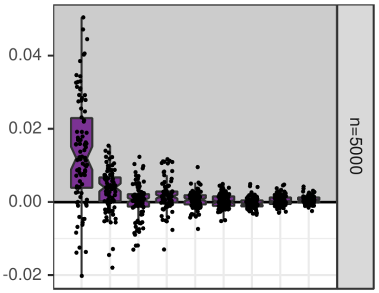


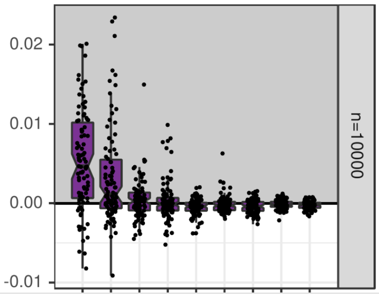
What is shown in the graphics?

- X axis: instances with an increasing tightness (from left to right)
- Y axis: improvement of CMSA over LNS (in percent)

MDKP Problem: CMSA vs. LNS

Larger instances





What is shown in the graphics?

- X axis: instances with an increasing tightness (from left to right)
- Y axis: improvement of CMSA over LNS (in percent)

Gabriela Ochoa
U. of Stirling

Katherine Malan
U. of S. Africa

ELSEVIER

Applied Soft Computing
Volume 109, September 2021, 107492

Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics

Gabriela Ochoa [a], Katherine M. Malan [b], Christian Blum [c]

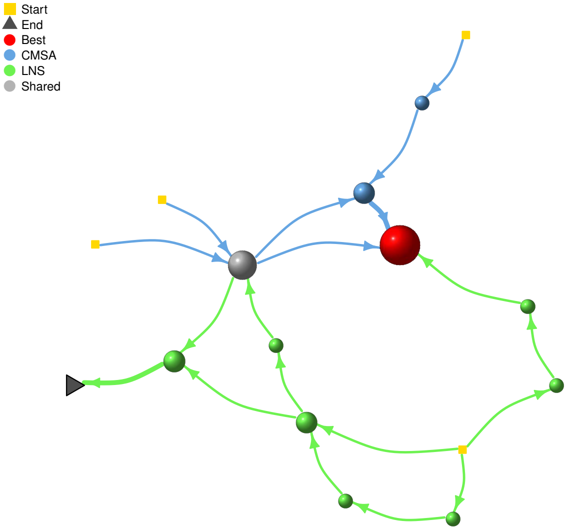
Available at:
`https://github.com/gabro8a/STNs.git`

STNWeb: Interactive STN application

Available at: `https://www.stn-analytics.com/`

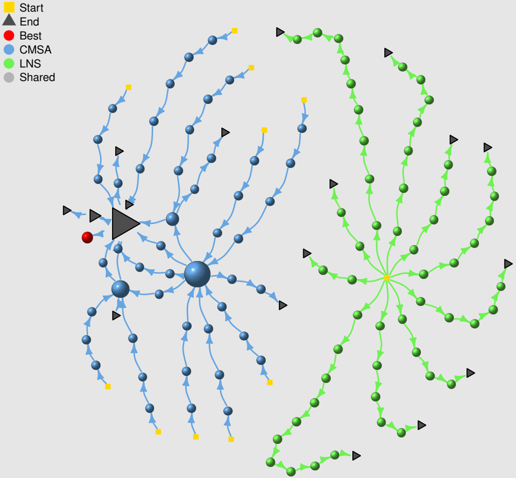
By: Camilo Chacón (IIIA)

STN Example Graphic



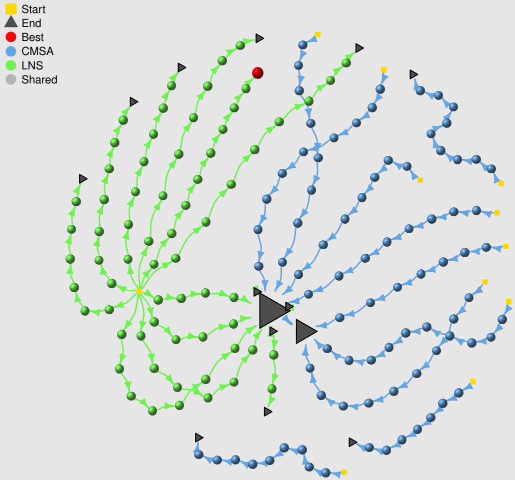
Visualization: CMSA vs. LNS

Problem instance with low *tightness* ($n = 10000$)



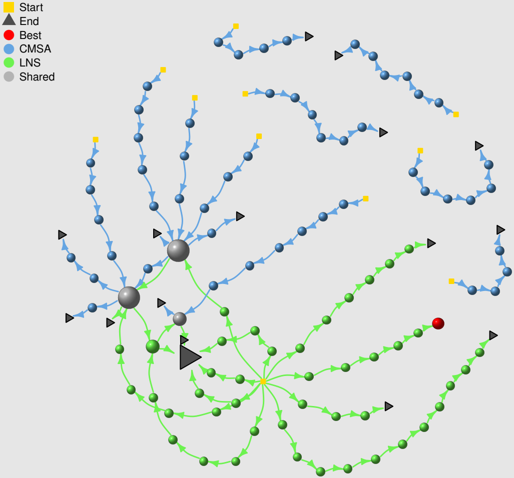
Visualization: CMSA vs. LNS

Problem instance with medium *tightness* ($n = 10000$)



Visualization: CMSA vs. LNS

Problem instance with high *tightness* ($n = 10000$)



CMSA: our own applications

Standard CMSA

- Minimum common string partition
- Several variants of the longest common subsequence problem
- Minimum covering arborescence
- Multi-dimensional knapsack
- Maximising the net present value of project schedules
- Minimum capacitated dominating set
- Route planning for cooperative air-ground robots with fuel constraints
- Maximum disjoint dominating sets
- Bus driver scheduling

Adapt-CMSA

- Minimum positive influence dominating set
- Different versions of electric VRPs
- Multi-way multi-dimensional number partitioning

CMSA: other researchers' work

- **[Hong et al., 2025]**: Flying Sidekick Traveling Salesman Problem, **CITA 2025**
- **[Martí et al., 2024]**: Bi-level diversity and equity optimization, **Journal of Heuristics**
- **[Ghirardi, Salassa, 2022]**: Maximum happy vertices problem, **Top, Springer**
- **[Ferrer et al., 2021]**: Prioritized pairwise test data generation in software product lines, **Journal of Heuristics**
- **[Dupin et al., 2021]**: Refueling and maintenance planning of nuclear power plants, **Journal of Heuristics**
- **[Hawa, 2020]**: Score-constrained Packing Problem, **Doctoral Thesis**
- **[Ben-Smida et al., 2019]**: Taxi sharing, **EuroCast 2019**

Outlook

Current and Future Work

- Replace (or bias) greedy information for solution construction with information learned employing machine learning techniques
- Can we use LLMs as assistants within CMSA
- Using state-of-the-art solvers within CMSA. Can we improve the performance of state-of-the-art solvers for well-studied problems by plugging them into CMSA? (very promising results for MaxSAT)

Group members, colleagues, and friends working on these tasks

M. A. Akbay

J. Reixach

M. Djukanovic

P. Pinacho

G. Rodríguez

Questions?