

First-improvement or best-improvement? An in-depth local search computational study to elucidate a dominance claim

D. Aloise, R. Moine, C. C. Ribeiro, J. Jalbert

G-2024-47

Août 2024

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Citation suggérée : D. Aloise, R. Moine, C. C. Ribeiro, J. Jalbert (August 2024). First-improvement or best-improvement? An in-depth local search computational study to elucidate a dominance claim. Rapport technique, Les Cahiers du GERAD G- 2024-47, GERAD, HEC Montréal, Canada.

Suggested citation: D. Aloise, R. Moine, C. C. Ribeiro, J. Jalbert (Août 2024). First-improvement or best-improvement? An in-depth local search computational study to elucidate a dominance claim, Technical report, Les Cahiers du GERAD G-2024-47, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (<https://www.gerad.ca/fr/papers/G-2024-47>) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Before citing this technical report, please visit our website (<https://www.gerad.ca/en/papers/G-2024-47>) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

First-improvement or best-improvement? An in-depth local search computational study to elucidate a dominance claim

Daniel Aloise ^{a, b}

Robin Moine ^{a, b}

Celso C. Ribeiro ^{c, d}

Jonathan Jalbert ^{e, b}

^a *Département de génie informatique et génie logiciel, Polytechnique Montréal, Montréal, (Qc), Canada, H3T 1J4*

^b *GERAD, Montréal (Qc), Canada, H3T 1J4*

^c *Universidade Federal Fluminense, Institute of Computing, Niterói, Rio de Janeiro, Brazil, 24210*

^d *Universidade Federal da Bahia, Institute of Computing, Niterói, Rio de Janeiro, Brazil, 24210*

^e *Département de mathématiques et de génie industriel, Polytechnique Montréal, Montréal, (Qc), Canada, H3T 1J4*

daniel.aloise@gerad.ca

j.jalbert@polymtl.caa

Août 2024

Les Cahiers du GERAD

G–2024–47

Copyright © 2024 Aloise, Moine, Ribeiro, Jalbert

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : Local search methods start from a feasible solution and improve it by successive minor modifications until a solution that cannot be further improved is encountered. They are a common component of most metaheuristics. Two fundamental local search strategies exist: first-improvement and best-improvement. In this work, we perform an in-depth computational study using consistent performance metrics and rigorous statistical tests on several classes of test problems considering different initialization strategies and neighborhood structures to evaluate whether one strategy is dominant over the other. The numerical results show that computational experiments previously reported in the literature claiming the dominance of one strategy over the other for the TSP can not be extrapolated to other problems. Still, our results reinforce the need for extensive experimentation to decide the most appropriate strategy for each specific problem and context since a rule of thumb does not seem to exist for deciding which local search strategy is the best in the general case.

Keywords : Local search, heuristics, first-improvement, best-improvement, combinatorial optimization

Acknowledgements: This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant 2023-04466; the Brazilian National Council for Scientific and Technological Development (CNPq), grant 309869/2020-0; and Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), grant E-26/200.926/2021.

1 Motivation

Combinatorial optimization involves finding optimal solutions to problems defined over a discrete set of feasible solutions. Any combinatorial optimization problem can be formulated as the constrained minimization (resp. maximization) of some function $f(x) : x \in F$. Here, E is a discrete ground set formed by the elements that form the problem solutions, $F \subseteq 2^E$ is the set of feasible solutions, and $f : 2^E \rightarrow \mathbb{R}$ is the objective function Papadimitriou and Steiglitz (1982). A subset of the elements of the ground set defines a solution x for this problem. We seek an *optimal solution* (or, simply, an *optimum*) $x^* \in F$ such that $f(x^*) \leq f(x)$ (resp. $f(x^*) \geq f(x)$), for all $x \in F$. The ground set E , the objective function f , and the feasible set F are specific to each problem instance.

Opposed to exact methods, *approximate methods* or *heuristics* provide feasible solutions that are not necessarily optimal. Approximate methods generally run faster than exact methods and can handle larger problem instances. *Local search* methods start from a feasible solution and improve it by successive minor modifications until a solution that cannot be further improved is encountered. Although they often provide high-quality solutions whose values are close to those of the optimal solutions, they can become prematurely trapped in low-quality, locally optimal solutions in some situations.

A local search can be seen as a partial traversal of the solution space in this setting. Local search methods start from any feasible solution and visit other (feasible or infeasible) solutions until a feasible solution is found that can not be further improved. Local improvements are evaluated concerning neighbor solutions that can be obtained by slight modifications applied to the current solution. Local search methods are a common component of most metaheuristics. Yagiura and Ibaraki (2002) traced the history of local search since the work of Croes (1958). Hoos and Stützle (2005) developed a thorough study of the foundations and applications of stochastic local search, i.e., methods based on local search that use randomization to generate or select candidate solutions for combinatorial optimization problems. In their seminal work, Lin and Kernighan (1973) developed a local search heuristic based on 2-opt and 3-opt exchanges for approximately solving the symmetric traveling salesman problem (TSP), one of the best approaches for the problem.

Two fundamental local search strategies are *first-improvement* and *best-improvement*. At any iteration of a *first-improvement* local search strategy, the algorithm moves from the current solution to any neighbor with a better (i.e., improving) value for the objective function. In the case of a *best-improvement* local search strategy, at any iteration, the algorithm moves from the current solution to its best neighbor whenever the latter improves the former. Detailed pseudo-codes for implementing these two fundamental strategies are presented in A. Given a problem instance and an initial solution, these two strategies will not necessarily converge to the same local optimum.

However, there is no consensus in the literature regarding which local search strategy performs better in a general role among first-improvement or best-improvement. Authors often compare first-improvement and best-improvement versions of an implemented local search regarding aggregated statistical metrics, such as the average and standard deviations of the obtained local optima, computed across several initializations and data instances of a particular optimization problem.

Hansen and Mladenović (2006) presented an empirical study comparing the performance of first-improvement and best-improvement strategies for the TSP using the 2-opt neighborhood. They performed tests on randomly generated Euclidean and non-Euclidean TSP instances and a subset of TSPLIB Reinelt (1991) instances. The main finding of their work is a *dominance claim* for the TSP summarized in its abstract and quoted below:

“When applying the 2-opt heuristic to the travelling salesman problem, selecting the best improvement at each iteration gives worse results on average than selecting the first improvement, if the initial solution is chosen at random. However, starting with ‘greedy’ or ‘nearest neighbor’ constructive heuristics, the best improvement is better...”

Their paper still provides conclusions on comparing the local search versions on other metrics than solution cost.

In this work, and based on the literature review presented in the next section, we show that this conclusion has been abusively extended to multiple contexts, many addressing other NP-hard problems.

In addition, we revisit the computational study of Hansen and Mladenović (2006) using more rigorous statistical tests, replicating this analysis to additional test problems to verify the above claim thoroughly. We show that different conclusions stand for some problems and neighborhoods when the strategy opposed to that pointed out by the claim prevails. In particular, we show that the conclusions of Hansen and Mladenović (2006) were biased using an inappropriate metric. We apply a more appropriate one in our computational study, leading to more consistent results.

The remainder of the paper is organized as follows. Section 2 tracks the impact of the dominance claim of Hansen and Mladenović (2006) in the specialized literature. Section 3 discusses the metric used in their computational experiments and proposes a more appropriate and consistent metric. Section 4 describes the setup of our computational experiments and statistical tests. The computational experiments performed in Section 5 show that although the dominance claim could be confirmed for the TSP using the 2-opt neighborhood, it was not confirmed for other classes of test problems. Consequently, it cannot be asserted that any of the best-improvement or first-improvement strategies generally prevails over the other based solely on the nature of the initialization. Concluding remarks are drawn in the last section.

2 Literature review

The article of Hansen and Mladenović (2006) has been very influential in the combinatorial optimization literature, with 158 citations according to Google Scholar as of January 31, 2024. Among these, 37 directly mention the dominance claim of Hansen and Mladenović (2006), which asserts the prevalence of one local search strategy over the other depending on initialization. Furthermore, many of these articles have arbitrarily extended the dominance claim to other problems beyond the TSP. However, the appropriation of this claim varies among the citing works. We classify these citations into three groups:

Group A Manuscripts that use the dominance claim to choose a local search version (first-improvement or best-improvement) without computational testing.

Group B Manuscripts that mention the dominance claim but perform their experiments to determine which local search strategy is better.

Group C Manuscripts that mention the dominance claim but do not present computational experiments concerning the two local search strategies.

We found 13 manuscripts in group *A*, 13 in group *B*, and 11 in group *C*. We discarded the remaining 121 manuscripts since they cited Hansen and Mladenović (2006) without mentioning the dominance claim. Nonetheless, it is noteworthy that they also refer the reader to that paper, drawing attention to the dominance claim under closer scrutiny here.

Table 1 details the broad spectrum of application domains influenced by Hansen and Mladenović (2006). These papers tackle varied NP-hard combinatorial optimization problems. Notably, a large parcel of the works in this table relate to routing problems akin to the TSP: many papers in groups *A* and *B* also use the 2-opt neighborhood to exploit the search space.

The dominance claim directly influences the manuscripts in the group *A*, and they do not even perform any comparative studies to check its validity. Papers in group *B* perform some comparative

Table 1: Addressed optimization problems by group.

Group	Application domain	
	2-opt neighborhood	Other neighborhoods
<i>A</i>	TSP and routing Dawid (2008); Tang (2008), location and distribution Lopes et al. (2016); Turan et al. (2017)	Scheduling Abderrahim et al. (2020), data mining Costa et al. (2017); Pereira et al. (2018); Pereira (2017) bin packing problems Goos et al. (2020); Sánchez-Oro et al. (2017), Portfolio optimization Akbay et al. (2020), and queuing systems Wood (2011)
<i>B</i>	TSP and routing Amaral et al. (2021); Babin et al. (2007); Becker et al. (2023); Irnich et al. (2006); Mjirda et al. (2016); Nascimento Silva et al. (2020)	Location Mladenović et al. (2019), data mining Mladenović et al. (2016), binary quadratic programming Drugan and Thierens (2012), graph theory Brimberg et al. (2009), project management Yassine et al. (2013), and scheduling Hackl (2018)
<i>C</i>		TSP and routing Almoustafa (2013); Bon-toux (2008); Erdelić and Carić (2019), location Brimberg and Salhi (2019), packing Rajab (2012), reinforcement learning da Costa et al. (2021), and scheduling Bougrine et al. (2017); Tari et al. (2021)

experiments, with results that sometimes contradict the dominance claim: four out of the 13 articles contradicted it, including two of them using the 2-opt neighborhood. Mjirda et al. (2016), in particular, pointed out that the claim in Hansen and Mladenović (2006) possibly does not hold for other problems or algorithms. However, a deeper investigation was never carried out, and many authors still take this claim for granted.

Finally, although the manuscripts in group *C* do not involve computational experiments concerning the dominance claim, they propagate the conclusion presented in Hansen and Mladenović (2006), potentially influencing other authors about the best local search strategy to choose. More than 2000 articles cite the manuscripts in groups *A*, *B*, and *C* according to Google Scholar as of January 2024.

3 Choice of a comparison metric

In this section, we discuss the choice of an appropriate metric to compare the relative performance of the first-improvement and best-improvement local search strategies when both use the same initial solution. We define the following notation:

- x_{init} : the initial solution used by both strategies.
- x_{BI} : a locally optimal solution found by the best-improvement strategy.
- x_{FI} : a locally optimal solution found by the first-improvement strategy.
- $f(x)$: the objective function value of solution x .

We first discuss the metric used by Hansen and Mladenović (2006), defined as

$$improv_1(x_{BI}, x_{FI}) = \frac{f(x_{BI}) - f(x_{FI})}{f(x_{FI})}, \quad (1)$$

which measures the difference (positive or negative) between the costs of the solutions obtained by the two strategies, normalized by the cost of the solution obtained by the first-improvement strategy. This value is then averaged in Hansen and Mladenović (2006) over the total number of runs in each particular experiment to claim that one strategy outperforms the other.

However, we argue that this normalization choice is rather arbitrary. Let us suppose that the improvement between x_{BI} and x_{FI} is computed instead as

$$\text{improv}'_1(x_{BI}, x_{FI}) = \frac{f(x_{BI}) - f(x_{FI})}{f(x_{BI})}, \quad (2)$$

i.e., with $f(x_{BI})$ replacing $f(x_{FI})$ in the denominator of equation (1) as the normalization factor.

We now suppose that first-improvement and best-improvement local search strategies were applied to five distinct initial solutions. The costs of the local minima obtained for each of the initial solutions are 5, 3, 4, 3, and 3 by first-improvement, and 4, 3, 4, 5, and 2 by best-improvement. The average values of improv_1 and improv'_1 computed over the five runs are +0.027 and -0.070, respectively. Thus, a positive value of improv_1 suggests that, on average, first-improvement outperforms best-improvement, while a negative value of improv'_1 indicates the opposite – i.e., on average, best-improvement outperforms first-improvement.

The arbitrariness of the denominator choice and the sensitivity of the conclusion to that choice indicate the inadequacy of this metric. Therefore, we use in our study another metric commonly used in the literature to assess the relative performance of the solutions obtained by local search methods, namely:

$$\text{improv}_2(x_{BI}, x_{FI}) = \frac{f(x_{BI}) - f(x_{FI})}{f(x_{init})}. \quad (3)$$

This metric gives the improvement (or deterioration) yielded by x_{BI} over x_{FI} concerning the cost of the initial solution. It is not arbitrary in the sense that neither of the solution strategies biases the normalization. Thus, the two strategies are compared on a standard basis, i.e., the cost of the same initial solution used for both.

4 Experimental setup

To further investigate the dominance claim in Hansen and Mladenović (2006), we devised an extended computational experiment and a deeper statistical analysis of the results obtained. These experiments consider not only the TSP as in Hansen and Mladenović (2006) but also three distinct NP-hard combinatorial problems: the weighted MAX-SAT (wMAX-SAT) Krentel (1988), the minimum sum of squares clustering (MSSC) Aloise et al. (2009), and the single-machine total-weighted tardiness (SMTWTP) Du and Leung (1990) problems. In this section, we present (i) a concise description of these three problems, as well as the neighborhoods explored in the associated local searches; (ii) the different tested initialization methods; (iii) the problem instances used in our study, and (iv) the description of the experiments and statistical tests performed to evaluate the results.

4.1 Problems and neighborhoods

4.1.1 Traveling salesman problem

Let $G = (V, E)$ be a graph with node set $V = \{1, \dots, n\}$ and edge set $E \subseteq V \times V$. A non-negative length $d_{i,j}$ is associated with each existing edge $(i, j) \in E$. A Hamiltonian cycle (or a tour) is a cycle that visits all nodes of G exactly once. The traveling salesman problem (TSP) involves finding a minimum length tour of G . The length (or cost) of a feasible TSP solution is computed simply by summing up the lengths of its edges.

The 2-opt neighborhood for the TSP is defined by replacing any pair of nonadjacent edges of a tour with the unique pair of edges that recreates a Hamiltonian cycle. Figure 1(a) illustrates a tour in a graph with $|V| = 8$ nodes. Figure 1(b) depicts one of the solutions in its 2-opt neighborhood, obtained by replacing the pair of edges (5,6) and (7,8) with the new edges (5,7) and (6,8). The number

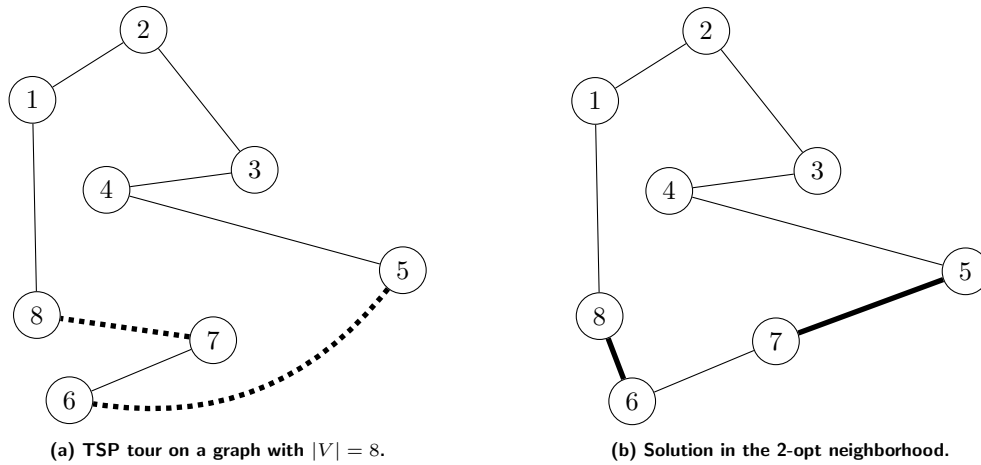


Figure 1: Illustration of a neighbor in the 2-opt neighborhood for the TSP.

of potential solutions in the 2-opt neighborhood amounts to $O(|V|^2)$, corresponding to the exchange of all possible pairs of edges.

In this work, we additionally explore the 3-opt neighborhood Lin (1965) for the TSP. This neighborhood is formed by all tours that can be obtained by removing any three edges from a tour, subsequently creating a new, different tour by adding three different edges that reconnect the tour. Figure 2(a) illustrates the same previously shown tour in a graph with $|V| = 8$ nodes. Figure 2(b) depicts one of the solutions in its 3-opt neighborhood, obtained by replacing edges (3,4), (5,6), and (7,8) with the new edges (3,5), (4,7), and (6,8). We note that the number of potential solutions in the 3-opt neighborhood increases to $O(|V|^3)$, corresponding to all possible choices of triples of edges to be removed.

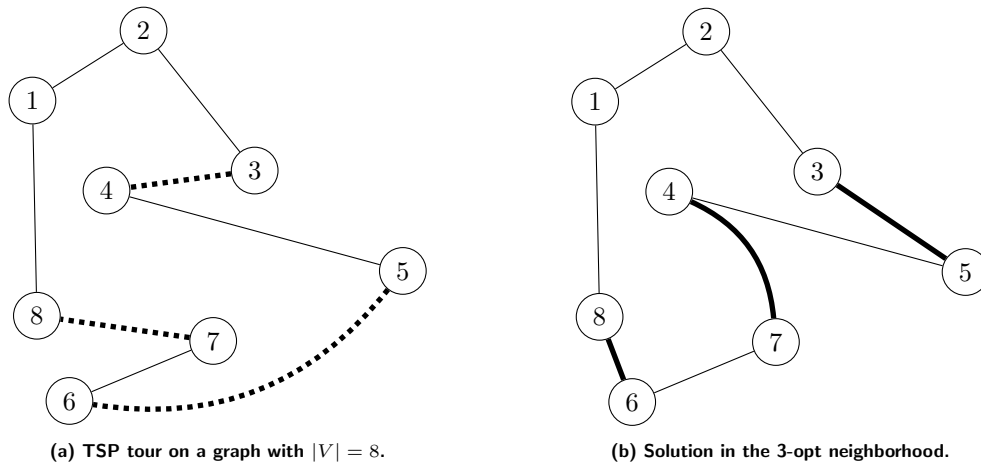


Figure 2: Illustration of a neighbor in the 3-opt neighborhood for the TSP.

4.1.2 Weighted MAX-SAT

Given the sets $X = \{x_1, \dots, x_{n_{SAT}}\}$ of boolean variables and $C = \{c_1, \dots, c_{m_{SAT}}\}$ of clauses, where each clause is a disjunction of literals (i.e., a variable or its complement), and weights w_i associated to each clause $c_i, i = 1, \dots, m_{SAT}$, the weighted MAX-SAT problem (wMAX-SAT) consists in finding an assignment of truth values to the variables in X such that the sum of the weights of the satisfied clauses is maximized.

In this work, the first-improvement and best-improvement local search strategies for wMAX-SAT are assessed with regard to exploring 1-opt neighbor solutions obtained by complementing the truth value of one single variable, either from true to false or from false to true. Figure 3(a) illustrates an instance of wMAX-SAT. Figure 3(b) exhibits a solution with a complete truth assignment, and Figure 3(c) shows a 1-opt neighbor solution obtained by complementing the value of x_1 .

$$\begin{matrix} (x_1 \vee x_2) & \wedge & (x_1 \vee \neg x_2) & \wedge & (x_2 \vee \neg x_3) & \wedge & (\neg x_1 \vee \neg x_2) \\ w_1 = 2 & & w_2 = 3 & & w_3 = 5 & & w_4 = 2 \end{matrix}$$

(a) wMAX-SAT instance.

$$(x_1 = 0, x_2 = 0, x_3 = 1) : w_2 + w_4 = 5$$

(b) wMAX-SAT solution.

$$(x_1 = 1, x_2 = 0, x_3 = 1) : w_1 + w_2 + w_4 = 7$$

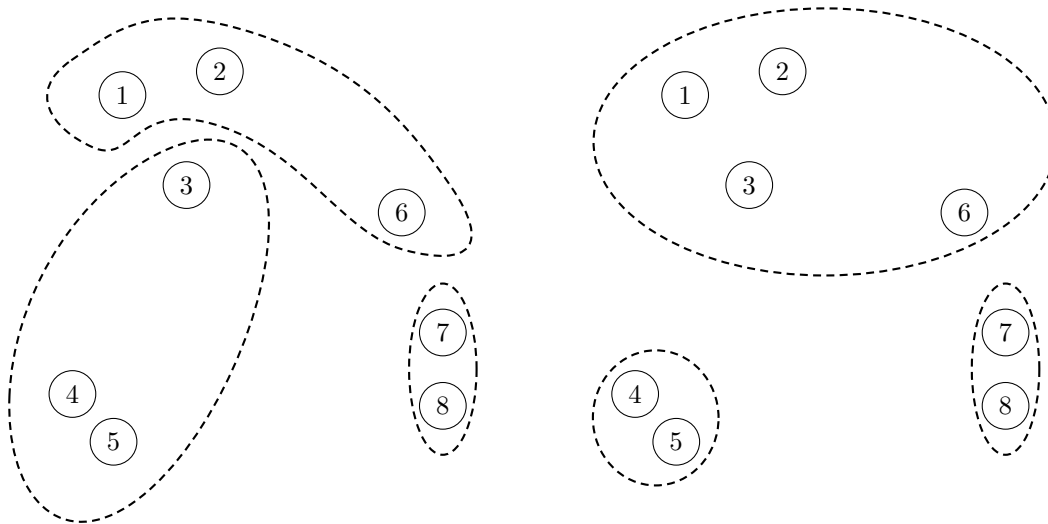
(c) Neighbor solution obtained by complementing the truth value of x_1 .

Figure 3: Illustration of a neighbor in the 1-opt neighborhood of wMAX-SAT.

4.1.3 Minimum sum of squares clustering

Clustering consists of partitioning a set P of data points into k subsets called clusters. In the minimum sum of squares clustering problem (MSSC), the objective is to find k clusters that minimize the sum of the squared Euclidean distances between each data point and its cluster centroid.

In this work, the analysis of the first-improvement and best-improvement local search strategies for MSSC considers the same swap neighborhood explored by the H -means heuristic (see, e.g., Pereira et al. (2018)). From a given MSSC solution, this neighborhood comprises all solutions obtained by changing the cluster membership of individual data points. An illustrative example of a neighbor solution in this neighborhood is shown in Figure 4, where point 3 changes its cluster membership.



(a) MSSC clustering solution.

(b) Solution in the H-means neighborhood.

Figure 4: Illustration of a neighbor in the H-means swap neighborhood of MSSC.

4.1.4 Single-machine total-weighted tardiness problem

The single-machine total-weighted tardiness problem (SMTWTP) considers the scheduling of a set $J = \{j_1, \dots, j_{n_J}\}$ of n_J independent jobs on one machine. Each job $j_i \in J$ is characterized by its processing time p_i , due date d_i , and weight w_i . Given a schedule ϕ of the jobs in J , the tardiness of job j_i is computed as $T_i = \max\{0, C_i - d_i\}$, where C_i refers to the completion time of job j_i in ϕ . The total weighted tardiness of ϕ , denoted $WT(\phi)$, is defined as $WT(\phi) = \sum_{i=1}^{n_J} w_i T_i$.

In this work, we evaluate first-improving and best-improvement local search strategies for SMTWTP within the exchange neighborhood Hoos and Stützle (2005), which encompasses neighbor solutions obtained by exchanging the positions of any two jobs in a given schedule. Figure 5(a) presents an instance of SMTWTP. Figure 5(b) illustrates an SMTWTP solution given by $\phi = \{j_2, j_3, j_4, j_1\}$, whereas Figure 5(c) illustrates one of its neighbor solutions $\phi' = \{j_2, j_1, j_4, j_3\}$ in the exchange neighborhood, obtained by switching the positions of jobs j_1 and j_3 in the schedule.

	j_1	j_2	j_3	j_4
p_i	3	5	2	4
d_i	6	8	5	7
w_i	2	3	1	2

(a) SMTWTP instance.

ϕ	j_2	j_3	j_4	j_1	$WT(\phi)$
C_i	5	7	11	14	
T_i	0	2	4	8	
$w_i T_i$	0	2	8	16	26

(b) SMTWTP solution.

ϕ'	j_2	j_1	j_4	j_3	$WT(\phi')$
C_i	5	8	12	14	
T_i	0	2	5	9	
$w_i T_i$	0	4	10	9	23

(c) Neighbor solution obtained by exchanging the positions of jobs j_1 and j_3 .

Figure 5: Illustration of a neighbor in the exchange neighborhood of SMTWTP.

4.2 Initialization methods

The dominance claim of Hansen and Mladenović (2006) states that one local search strategy (first-improvement or best-improvement) outperforms the other depending on how they are initialized. As such, our computational experiments were performed, starting the search with random and greedy solutions and assessing if that decision impacts the performance of the evaluated local search strategies.

We summarize below these two initialization approaches for each of the four problems considered in this work and described in Section 4.1. Detailed pseudo-codes and codes in C++ for all initialization methods are available in Moine (2024).

4.2.1 Traveling salesman problem

The random initialization sorts the n nodes at random, and outputs the TSP tour obtained by connecting the nodes in that order. The greedy method is the popular nearest neighbor heuristic for the TSP Laporte (1992); Lawler et al. (1985); Rosenkrantz et al. (1977). It begins with a randomly selected node, and adds the closest unvisited node to it. The latter becomes the incumbent and the previous step is repeated, until all nodes are visited. The TSP tour is completed by returning to the initial node.

4.2.2 Weighted MAX-SAT

Random initialization for wMAX-SAT generates a solution by assigning truth values (true or false) to the variables at random. The greedy initialization selects a variable to assign a truth value at each iteration. The selected variable is the one that, after its truth value assignment (to either true or false), maximizes the total weight of the yet-unsatisfied clauses that become satisfied.

4.2.3 Minimum sum of squares clustering

The random initialization generates a solution to MSSC by randomly assigning each data point to a cluster. The greedy initialization is derived from the k -means++ heuristic of Arthur and Vassilvitskii (2006). In our adapted method, the initial centroid is randomly chosen from the data points in set P , and the other $k - 1$ centroids are selected iteratively. The method chooses the next centroid as the farthest data point from its closest centroid among those already selected. Once all centroids are determined, the remaining $|P| - k$ data points are assigned to their nearest centroids.

4.2.4 Single-machine total-weighted tardiness problem

Random initial solutions for SMTWTP are obtained by random permutations of the jobs in J . Greedy solutions are constructed by using the Modified Due Date (MDD) heuristic Hoos and Stützle (2005); Bauer et al. (1999). This heuristic sequences jobs in ascending order of their modified due dates, calculated as $mdd_j = \max\{C + p_j, d_j\}$, where C represents the cumulative processing time of the previously scheduled jobs in the partial solution.

4.3 Problem instances

We conducted our experiments on random instances of the four classes of test problems.

4.3.1 Traveling salesman problem

We followed the same scheme used in Hansen and Mladenović (2006) to generate the random TSP instances. Nodes of the graph $G = (V, E)$ were uniformly selected from a 100×100 square, with the number of nodes $|V| \in \{20, 30, \dots, 150\} \cup \{200, 250, \dots, 500\} \cup \{500, 600, \dots, 1000\}$. For each value of $|V|$, 1000 instances were generated.

4.3.2 Weighted MAX-SAT

We generated random weighted MAX-SAT instances for $n_{SAT} \in \{50, 60, \dots, 90\} \cup \{100, 150, 200\}$ and $m_{SAT} \in \{1000, 1200, \dots, 2000\}$, with clause integer weights sampled from the uniform distribution in the interval $[1, max_w]$, where $max_w \in \{50, 100\}$. The maximum number of literals per clause was limited to three. For each combination of n_{SAT} , m_{SAT} and max_w , we generated 1000 distinct instances.

4.3.3 Minimum sum-of-squares clustering

We randomly generated MSSC instances by sampling points from a 100×100 square. For each number of points $|P| \in \{20, 30, \dots, 150\} \cup \{200, 250, \dots, 500\} \cup \{500, 600, \dots, 1000\}$, instances with $k \in \{2, 4, 8, 16, 32, 64, 128, 256\}$ clusters were created. Then, for each combination of $|P|$ and k , 1000 instances were generated. We simulated the clusters in the 100×100 square using the bivariate Gaussian distribution. Let $(\mu_i : i = 1, \dots, k)$ denote the k bidimensional mean vectors uniformly sampled on the 100×100 square. Clusters were generated using Gaussian distributions $\mathcal{N}(\mu_i, \sigma^2 I_2)$, where σ^2

is chosen from the interval $(100, 250)$. When σ^2 is small, clusters are well-defined; when σ^2 is large, clusters are much more diffuse.

4.3.4 Single-machine total-weighted tardiness problem

We created random SMTWTP instances using the generation scheme proposed in Potts and Van Wassenhove (1985) for $n_J \in \{40, 50, \dots, 100\}$. For every job j_i , an integer processing time p_i was sampled from a uniform distribution in $[1, 100]$, and an integer processing weight w_i was generated uniformly from $[1, 10]$. Then, job due dates, for $i = 1, \dots, n_J$, were generated using uniform distributions defined over different parametrized ranges. For a relative range of due dates $rdd \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and a given average tardiness factor $tf \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, an integer due date d_i is sampled from the uniform distribution in the interval $[z \times (1 - tf - rdd/2), z \times (1 - tf + rdd/2)]$, where $z = \sum_{i=1}^{n_J} p_i$. A total of 1000 distinct instances were generated for each combination of n_J , rdd and tf .

4.4 Description of the experiments

The finding of Hansen and Mladenović (2006) that resulted in their dominance claim presented in Section 1 for the TSP (with the 2-opt neighborhood used for local search) suggested that the best-improvement strategy yields better local optima compared to first-improvement when greedy initializations are employed. Conversely, they claimed that the first-improvement strategy obtained better solutions when random initializations were used. We have shown in Section 3 that the results on which these claims were based have been computed using the inappropriate metric $improv_1$ (1) that could lead to biased conclusions.

To investigate and more accurately assess the existence (or not) of a significant difference (and to what extent) between the quality of the local optima obtained by the first-improvement and best-improvement strategies depending on the initialization method, we decided to perform deeper and more extended experiments, applying more rigorous statistical tests using the $improv_2$ metric (3). We were motivated to obtain more conclusive results and shed more light on this relevant experimental subject.

Each experiment performed and reported in this work refers to (i) a specific test problem (TSP, wMAX-SAT, MSSC, and SMTWTP), and (ii) a specific initialization method (random or greedy). In the experiments with the TSP, we also considered two different neighborhoods: 2-opt and 3-opt. In what follows, we describe the organization of the experiments.

4.4.1 Traveling salesman problem

A statistical test is performed for each $|V| \in \{20, 30, \dots, 150\} \cup \{200, 250, \dots, 500\} \cup \{500, 600, \dots, 1000\}$. Each test considers the results of the two local search strategies to compute the $improv_2$ metric for each of the 1000 random instances with the same number of nodes. Therefore, each experiment encompasses 27 statistical tests, one for each $|V|$ value.

4.4.2 Weighted MAX-SAT

A statistical test is performed for each combination of $n_{SAT} \in \{50, 60, \dots, 90\} \cup \{100, 150, 200\}$, $m_{SAT} \in \{1000, 1200, \dots, 2000\}$, and $max_w \in \{50, 100\}$. Thus, each experiment encompasses $8 \times 6 \times 2 = 96$ statistical tests, one for each combination of the above values.

4.4.3 Minimum sum of squares clustering

A statistical test is performed for each pair of values of $|P|$ and k , with $|P| \in \{20, 30, \dots, 150\} \cup \{200, 250, \dots, 500\} \cup \{500, 600, \dots, 1000\}$, and $k = 2^p$, with $p = \{1, 2, \dots, 8\}$. Here, an experiment encompasses $27 \times 8 = 216$ statistical tests, one for each combination of $|P|$ and k .

4.4.4 Single-machine total-weighted tardiness problem

A statistical test is performed for each combination of values of $n_J \in \{40, 50, \dots, 100\}$, $rdd \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $tf \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. As such, an experiment encompasses $7 \times 5 \times 5 = 175$ statistical tests.

4.4.5 Statistical tests

We have shown in the previous sections that a statistical test is performed for a sample of values of the $improv_2$ metric obtained from each experiment with randomly generated instances. We want to test the two following hypotheses:

$$\begin{cases} \mathcal{H}_0 : improv_2 = 0, \\ \mathcal{H}_1 : improv_2 \neq 0; \end{cases}$$

where the null hypothesis \mathcal{H}_0 represents that the two local search strategies are equivalent, while the alternative hypothesis \mathcal{H}_1 represents the opposite scenario. To test these hypotheses, the non-parametric Wilcoxon signed-rank test (Wilcoxon, 1945) is employed. A p -value smaller than 5% indicates a significant difference between the results obtained by the two local search strategies, while a larger p -value does not provide conclusive evidence.

In addition to the p -value, the effect size of a test statistic can be computed to indicate how large the observed effect is compared to random noise. In very large samples, the p -values can be confounded due to their dependence on both the sample size and the effect size, whereas the effect size remains independent of the sample size. In the Wilcoxon signed-rank test, the effect size can be computed by dividing the observed test statistic by the square root of the sample size. According to Cohen (2013), the effect size r for the Wilcoxon test can be categorized as in Table 2.

Table 2: Effect size r for the Wilcoxon test.

r	Effect size
$r < 0.1$	no effect
$0.1 \leq r < 0.3$	small effect
$0.3 \leq r < 0.5$	medium effect
$r \geq 0.5$	large effect

5 Results

In this section, we present the results of the experiments with the first-improvement and best-improvement local search strategies applied to the problems described in Section 4.1.

We used tailored pie charts to present the experimental results and facilitate their interpretation. Each pie chart is associated with one of the experiments reported in Section 4.4. In these experiments, the first-improvement and best-improvement strategies are assessed according to (i) the test problem (TSP, wMAX-SAT, MSSC, and SMTWTP), and (ii) the initialization method: random or greedy.

Thus, there will be pie charts associated with different combinations of the test problems and initialization methods. In addition, for the TSP, we also show results for two experiments with the 3-opt neighborhood.

Each pie chart is formed by at most three crown sectors. The green sector represents the proportion of statistical tests (with p -value < 0.05) where the first-improvement strategy outperformed the best-improvement strategy local search. Contrarily, the red sector indicates the proportion of the statistical tests where best-improvement prevailed over first-improvement. We recall that the dominance claim Hansen and Mladenović (2006) states that *first-improvement performs better for random initial solutions, while best-improvement performs better for greedy initializations*. Finally, the blue sector (denoted by NC) indicates the fraction of non-conclusive cases, when the p -value of the used statistical test is greater than or equal to 0.05, or when both local search strategies yield the same final local optimum value, i.e., $improv_2 = 0$, and consequently a statistical test cannot be performed.

The first two sector types, green and red, are further divided into two parts. Capital letters FI and BI denote strong dominance of first-improvement and best-improvement, respectively, indicating the proportion of statistical tests with medium and large effect sizes. Conversely, lowercase letters fi and bi represent weak dominance of first-improvement and best-improvement, respectively, corresponding to statistical tests exhibiting small effect sizes.

5.1 Traveling salesman problem experiments

5.1.1 Experiments with the 2-opt neighborhood

We first present the results obtained for the experiments with random instances of the TSP using the 2-opt neighborhood for local search. These were the main results used in Hansen and Mladenović (2006) to support the dominance claim, i.e., they broadly supported the claim that the first-improvement strategy performed better than best-improvement when the local search was initialized from a random solution and that the best-improvement strategy was preferred when initial solutions were given by a greedy method (i.e., nearest-neighbor).

We obtained very similar conclusions for the TSP from the experiments with random and greedy initializations up to this point, as illustrated in Figures 6(a) and 6(b). For the case of random initialization, 92.31% of the statistical tests confirmed that first-improvement outperforms best-improvement, in accordance to the dominance claim (80.77% with medium to large effect size), whereas, for the greedy initialization, the claim was confirmed by 100% of the statistical tests since best-improvement was always the prevailing strategy.

5.1.2 Experiments with the 3-opt neighborhood

This experiment aims to assess the effect of using a different, larger neighborhood for local search exploration in the case of the TSP. We use the same random instances considered in the experiment with the 2-opt neighborhood.

Figure 7(a) shows that the change of neighborhood from 2-opt to 3-opt makes the dominance of the first-improvement strategy more fragile when the random initialization method is used. Although the experiment analyzed in Figures 6(a) showed that the first-improvement strategy outperformed the best-improvement strategy for 92.31% of the statistical tests with the 2-opt neighborhood, this number decreased to only 53.85% when the 3-opt neighborhood was used. However, the neighborhood change did not have any effect in the outcome of the experiments with the greedy initialization, as shown in Figure 7(b).

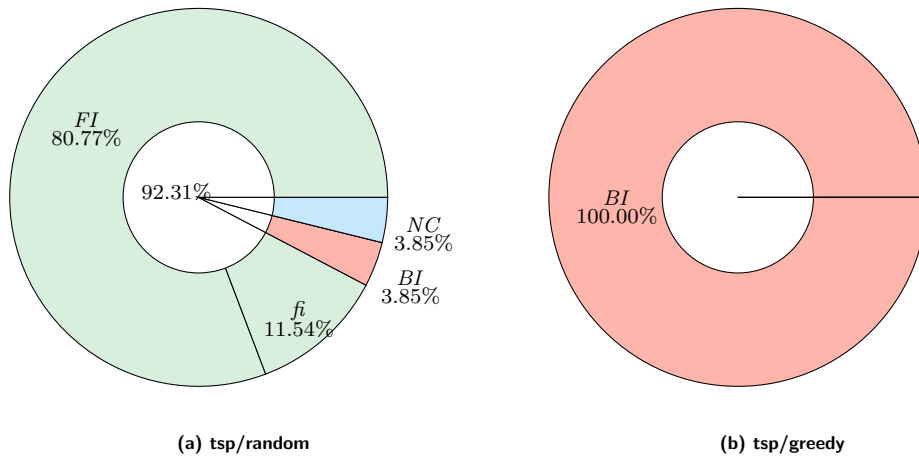


Figure 6: Pie-charts for the TSP using the 2-opt neighborhood on random instances with the random and greedy initialization methods.

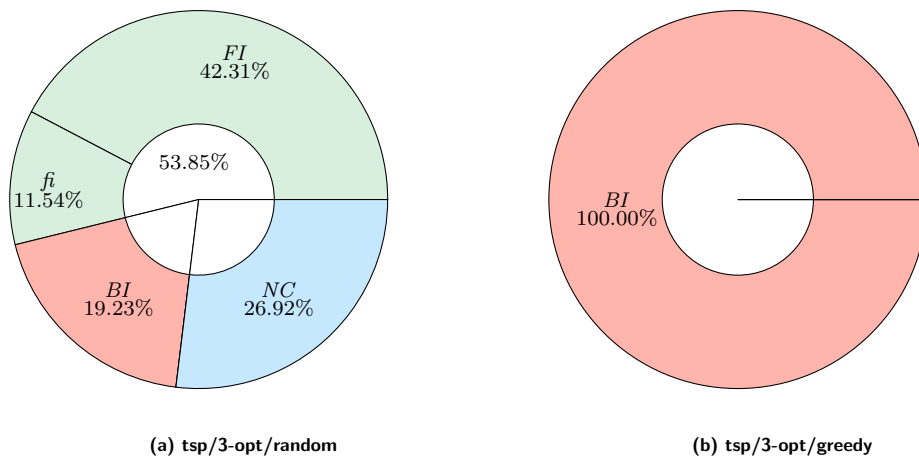


Figure 7: Pie-charts for the TSP using the 3-opt neighborhood on random instances with the random and greedy initialization methods.

In conclusion, we observe in this section that the dominance claim of Hansen and Mladenović (2006) was confirmed for the experiments with the TSP, not only with the 2-opt neighborhood considered in the original work, but also with the 3-opt neighborhood.

5.2 Weighted MAX-SAT experiments

In this section and the next two, we present results from experiments on three combinatorial optimization problems not addressed in the original work of Hansen and Mladenović (2006). Our goal here is to verify the dominance claim when applied to other combinatorial problems.

Figures 8(a) and 8(b) present results for the experiments on wMAX-SAT random instances. The results in Figure 8(a) shows that according to $FI = 87.50\%$ of the statistical tests, the random initialization yields better local minima when the first-improvement strategy is used, compared to best-improvement. Similarly, the results shown in Figure 8(b) shows that, with greedy initialization,

the best-improvement strategy outperforms the first-improvement strategy in 75% of the statistical tests. These results are aligned with the dominance claim of Hansen and Mladenović (2006).

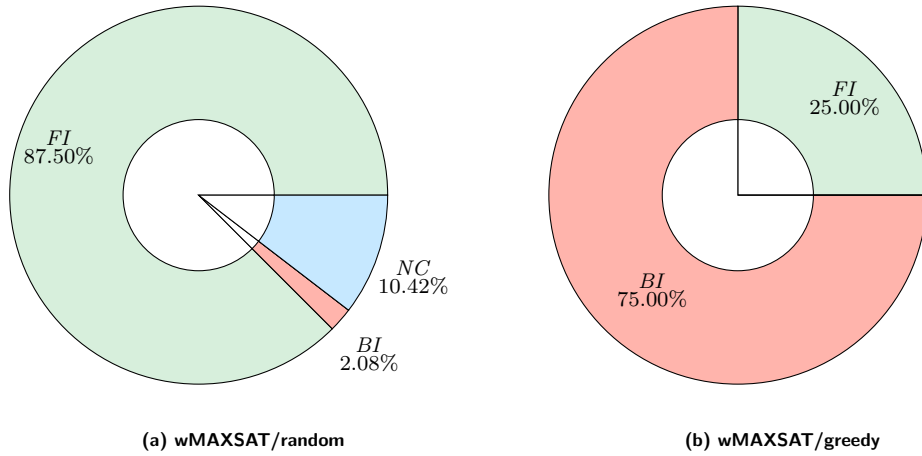


Figure 8: Pie-charts for wMAX-SAT problem on random instances with the random and greedy initialization methods.

5.3 Minimum sum of squares clustering experiments

Figures 9(a) and 9(b) present results for the experiments on MSSC random instances generated with Gaussian distributions using the random and greedy initialization methods. The results in Figure 9(a) for random initializations appear to confirm the dominance claim with $FI + fi = 86.21\%$. Contrarily, when greedy initialization is used, the dominance claim is not confirmed as observed in Figure 9(b) where $BI + bi = 35.06\%$, thus revealing that the first-improvement strategy prevails for greedy initialization as well.

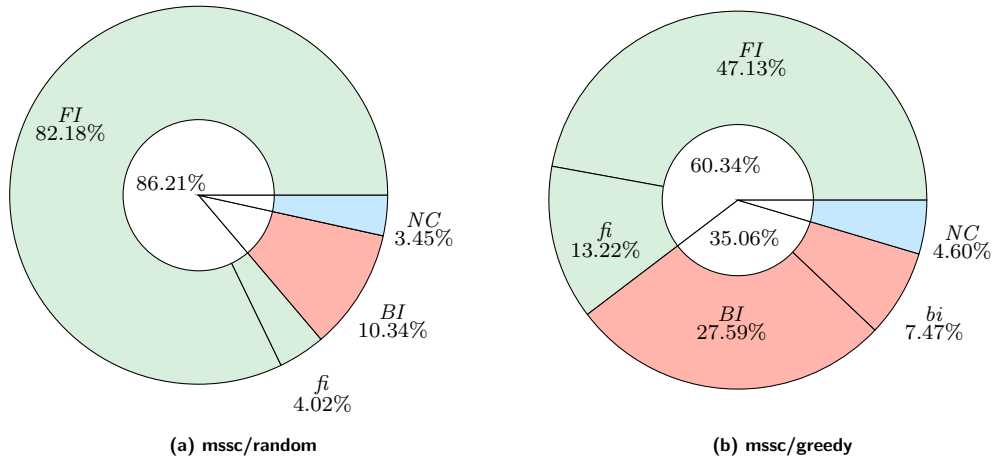


Figure 9: Pie-charts for MSSC problem on random instances with Gaussian distributed points with the random and greedy initialization methods.

5.4 Single-machine total weighted tardiness problem experiments

Figures 10(a) and 10(b) exhibit results for the experiments on SMTWTP random instances. The results in Figure 10(a) for random initialization contradict the dominance claim with $BI = 71.43\%$,

revealing that best-improvement outperforms first-improvement in this case. For greedy initialization, the results in Figure 10(b) confirm the dominance claim by a small margin with $BI = 45.71\%$ versus $FI + fi = 40.57\%$, and $NC = 13.71\%$.

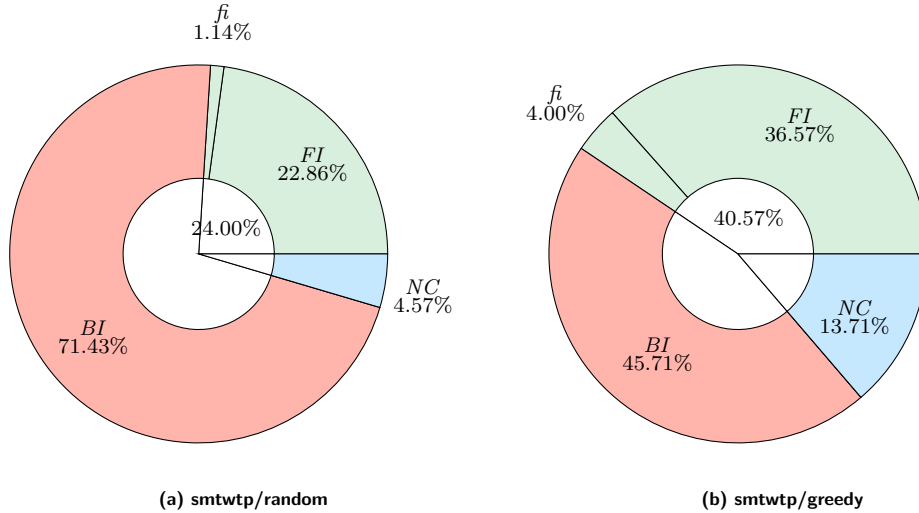


Figure 10: Pie-charts for the SMTWTP problem with the random and greedy initialization methods.

5.5 Summary of results

Table 3 summarizes the results of our experiments. They required a total of $27 + 96 + 216 + 175 = 514$ statistical tests, computed after the execution of $514 \times 1000 \times 2 = 1,028,000$ initialization methods (random and greedy), and $1,028,000 \times 2 = 2,056,000$ local searches (first-improvement and best-improvement versions).

Table 3: Summary of the experimental results.

Problem	Initialization	Best local search	Dominance claim
TSP	random	first-improvement	confirmed
	greedy	best-improvement	confirmed
wMAX-SAT	random	first-improvement	confirmed
	greedy	best-improvement	confirmed
MSSC	random	first-improvement	confirmed
	greedy	first-improvement	refuted
SMTWTP	random	best-improvement	refuted
	greedy	best-improvement	confirmed

At this point, a question arises: what happens in the case of initial solutions produced by *greedy randomized* heuristics? Such heuristics are widely used in metaheuristics as starting points for local search procedures (see, e.g., Resende and Ribeiro (2016)). In that case, when one local search strategy prevails over the other for both the random and greedy initializations (as for the MSSC and SMTWTP problems), the greedy randomized variant is not expected to change the conclusion about the prevailing local search strategy. However, the question arises when the prevailing strategy varies depending on the initialization (as for the TSP and wMAX-SAT problems). For such situations, we recommend that the users perform specific experiments for the greedy randomized initialization to verify which local search strategy is the best regarding the quality of the obtained local optima. The results presented in B show that, for the specific cases of the TSP and wMAX-SAT problems, the first-improvement strategy outperforms best-improvement when greedy randomized initializations are used.

6 Concluding remarks

In this work, we have revisited the work of Hansen and Mladenović (2006) on comparing first-improvement and best-improvement local search strategies as components of local search methods for the TSP using the 2-opt neighborhood. In particular, we addressed the validity of a dominance claim raised in this article that states the best strategy is determined by the type of initialization method used. It asserts that first-improvement should be preferred when starting the local search from random initial solutions, while a best-improvement strategy should be employed when starting the search from solutions produced by the nearest-neighbor greedy heuristic.

The work presented in Hansen and Mladenović (2006) has been highly influential in the literature, with the dominance claim being recklessly extrapolated to various other NP-hard combinatorial optimization problems beyond the TSP. Here, through an extensive computational study supported by rigorous statistical tests, we demonstrated that extrapolating this dominance claim to other classes of problems is not recommended, as it might lead to less effective local searches.

The primary step of our methodology consisted in showing that the metric used in Hansen and Mladenović (2006) to compare the performance of the two local search strategies was inappropriate due to the arbitrariness of its normalization. Nonetheless, correcting this metric did not alter the conclusions presented in Hansen and Mladenović (2006), as the dominance claim was confirmed by our computational experiments for the TSP under the 2-opt neighborhood.

However, our numerical experiments with three other test problems revealed that the dominance claim leads to the wrong choice of local search strategy for the minimum sum of squares clustering problem when the search is initiated from solutions constructed by a greedy heuristic, and for the single-machine total-weighted tardiness problem when the local search starts from random initial solutions.

Our recommendation is that users and researchers must perform specific experiments on local search strategies using appropriated metrics and statistical significance tests. Based on our computational experiments and observations, we believe that a rule of thumb does not exist for deciding which local search strategy is the best.

Besides the initialization method, we believe that the choice of neighborhood may also impact that decision. For example, our experiments with local search strategies for the TSP using the 3-opt neighborhood revealed that the dominance claim was more fragile in the case of random initialization than when the 2-opt neighborhood was used.

Last but not least, it is important to note that the quality of the local optima is only one of the criteria to consider when deciding on a local search strategy. Indeed several other criteria may co-exist such as the total computation time, or the robustness across different problem instances. Again, we believe there are no shortcuts when evaluating these criteria. For example, first-improvement iterations are typically faster since the search proceeds as soon as a better solution is found, but this can result in a large number of iterations before reaching a local optimum.

A Fundamental local search strategies

In what follows, we discuss the first-improvement and best-improvement strategies for implementing the neighborhood search.

At any iteration of an *iterative improvement* or *first-improvement* neighborhood search strategy, the algorithm moves from the current solution to any neighbor with a better (i.e., improving) value for the objective function. The new solution is the first improving solution identified along the neigh-

neighborhood search. The pseudo-code in Algorithm 1 describes a local search procedure based on the first-improvement strategy for a minimization problem. The search starts from a given initial solution x_{init} . A flag initialized in line 1 indicates whether or not an improving solution was found. The loop in lines 2 to 10 executes until replacing the current solution with a better neighbor becomes impossible. The flag is reset to `.FALSE.` in line 3 at the beginning of a new iteration. The loop in lines 4 to 9 visits every neighbor $x' \in N(x)$ of the current solution x until an improving solution is found. If the test in line 5 detects that the neighbor x' is better than the current solution x , then the latter is updated in line 6. Furthermore, the flag is reset to `.TRUE.` in line 7, indicating that a better solution was found, and a new iteration resumes. The algorithm returns the locally optimal solution x in line 11.

Algorithm 1: First-improvement local search for minimization

```

Input: Initial solution  $x_{init}$ 
Output: Locally optimal solution  $x_{FI}$ 
1 improvement  $\leftarrow$  .TRUE.;
2 while improvement = .TRUE. do
3   improvement  $\leftarrow$  .FALSE.;
4   for every  $x' \in N(x_{init})$  while improvement = .FALSE. do
5     if  $f(x') < f(x_{init})$  then
6        $x_{init} \leftarrow x'$ ;
7       improvement  $\leftarrow$  .TRUE.;
8     end
9   end
10 end
11 return solution  $x_{FI} = x_{init}$ .

```

In the case of a *best-improvement* local search strategy, at any iteration, the algorithm moves from the current solution to its best neighbor whenever the latter improves the former. The pseudo-code in Algorithm 2 describes a local search procedure based on the best-improvement strategy for a minimization problem. Once again, the search starts from any given initial solution x_{init} . A flag initialized in line 1 indicates whether or not an improving solution was found. The loop in lines 2 to 15 executes until replacing the current solution with a better neighbor becomes impossible. The flag is reset to `.FALSE.` in line 3 at the beginning of a new iteration. Variable f_{best} that stores the best objective function value over all neighbors of the current solution S is set to a large value in line 4. The loop in lines 5 to 10 visits every neighbor $S' \in N(S)$ of the current solution S . If the test in line 6 detects that the neighbor S' is better than the current best neighbor, then the latter is replaced with x' in line 7, and the best objective function value f_{best} in the neighborhood is updated in line 8. In line 11, we compare the current solution x with its best neighbor x_{best} . If f_{best} is less than $f(x)$, then the current solution is updated in line 12, the flag is reset to `.TRUE.` in line 13, indicating that a better solution was found, and a new iteration resumes. The algorithm returns the local optimum x in line 16. We observe that, independently of the starting solution and the neighborhood search strategy, the local search always stops at a local optimum. The complexity of each neighborhood search iteration depends mainly on two factors. First, it depends on the number of neighbors of each visited solution. Second, on the efficiency of the computation of the cost function value for each neighbor. Efficient implementations of the neighborhood search usually compute the cost of each neighbor S' by updating the cost of the current solution S , instead of calculating it from scratch and avoiding repetitive and unnecessary calculations.

Some ingenious implementation tricks, such as the use of candidate lists in best-improving strategies (to reduce the number of moves evaluated in each neighborhood search) or circular search in first-improving strategies (to avoid the reevaluation of non-improving moves already evaluated in the previous neighborhood search) can further improve the efficiency and the effectiveness of local search methods.

Algorithm 2: Best-improvement local search for minimization

```

Input: Initial solution  $x_{init}$ 
Output: Locally optimal solution  $x_{BI}$ 
1  $improvement \leftarrow .TRUE.$ ;
2 while  $improvement = .TRUE.$  do
3    $improvement \leftarrow .FALSE.$ ;
4    $f_{best} \leftarrow \infty$ ;
5   for every  $x' \in N(x_{init})$  do
6     if  $f(x') < f_{best}$  then
7        $x_{best} \leftarrow x'$ ;
8        $f_{best} \leftarrow f(x')$ ;
9     end
10  end
11  if  $f_{best} < f(x_{init})$  then
12     $x_{init} \leftarrow x_{best}$ ;
13     $improvement \leftarrow .TRUE.$ ;
14  end
15 end
16 return solution  $x_{BI} = x_{init}$ .

```

B Experiments with greedy randomized initialization

The experiments below refer to the comparison of first-improvement and best-improvement local searches when initiated from solutions obtained by greedy randomized heuristics. These heuristics introduce randomness into the greedy decision-making process by utilizing a weighted probability distribution.

The experiments presented here refer to the TSP (with the 2-opt neighborhood) and the wMAX-SAT. For these problems, the dominant local search strategy changed from first-improvement to best-improvement when the type of initialization used changed from random to greedy (see Table 3).

The implemented greedy randomized initialization method for the TSP uses a weighted probability distribution to select the next node. The probability of visiting an unvisited node follows the roulette wheel rule, i.e., it is inversely proportional to the distance to the last visited node. In the case of the wMAX-SAT, at each iteration, the greedy randomized initialization selects a variable to assign a truth value. The probability of choosing each variable is proportional to the total weight of the yet-unsatisfied clauses that would become satisfied after assigning it a truth value (either true or false). Also, in this case, the selection process is performed using the roulette wheel scheme.

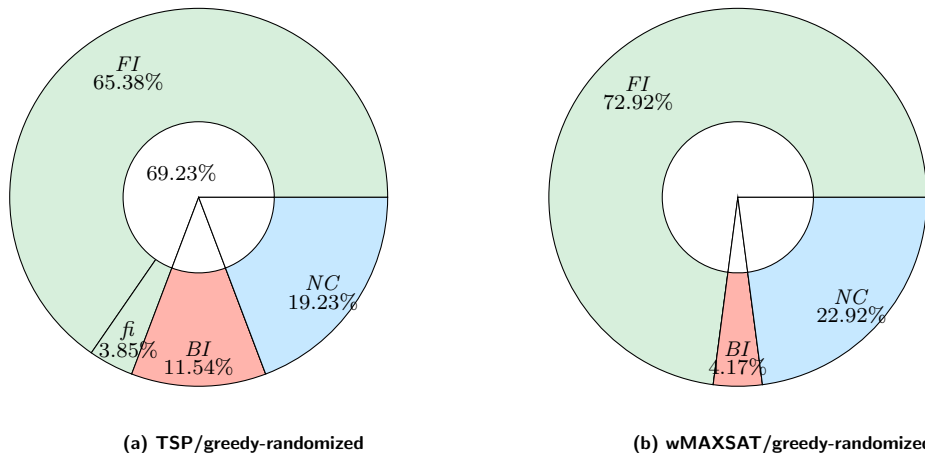


Figure 11: Pie-charts for the TSP and wMAXSAT problems with the greedy randomized initialization.

Figures 11a and 11b show results for the experiments on TSP and wMAX-SAT random instances, respectively, using the greedy randomized initialization. In both cases, the first-improvement strategy outperforms the best-improvement one regarding the quality of the local optima obtained. However, we observe that the prevalence of the first-improvement strategy over the best-improvement strategy becomes less pronounced when compared to that observed in Figures 6(a) and 8(b) regarding the pure random initialization, where $FI + fi = 92.31\%$ and $FI + fi = 87.50\%$, respectively.

References

- Abderrahim, M., Bekrar, A., Trentesaux, D., Aissani, N., Bouamrane, K., 2020. Bi-local search based variable neighborhood search for job-shop scheduling problem with transport constraints. *Optimization Letters* 16, 255–280.
- Akbay, M.A., Kalayci, C.B., Polat, O., 2020. A parallel variable neighborhood search algorithm with quadratic programming for cardinality constrained portfolio optimization. *Knowledge-Based Systems* 198, 105944.
- Almoustafa, S., 2013. Distance-constrained vehicle routing problem: exact and approximate solution (mathematical programming). Ph.D. thesis. Brunel University, School of Information Systems, Computing and Mathematics.
- Aloise, D., Deshpande, A., Hansen, P., Popat, P., 2009. NP-hardness of Euclidean sum-of-squares clustering. *Machine learning* 75, 245–248.
- Amaral, H.F., Urrutia, S., Hvattum, L.M., 2021. Delayed improvement local search. *Journal of Heuristics* 27, 923–950.
- Arthur, D., Vassilvitskii, S., 2006. k-means++: The advantages of careful seeding. Technical Report. Stanford University.
- Babin, G., Deneault, S., Laporte, G., 2007. Improvements to the Or-opt heuristic for the symmetric travelling salesman problem. *Journal of the Operational Research Society* 58, 402–407.
- Bauer, A., Bullnheimer, B., Hartl, R.F., Strauss, C., 1999. An ant colony optimization approach for the single machine total tardiness problem, in: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), IEEE. pp. 1445–1450.
- Becker, C., Gauthier, J.B., Gschwind, T., Schneider, M., 2023. In-depth analysis of granular local search for capacitated vehicle routing. *Discrete Applied Mathematics* 329, 61–86.
- Bontoux, B., 2008. Techniques hybrides de recherche exacte et approchée: application à des problèmes de transport. Ph.D. thesis. Université d'Avignon et des Pays de Vaucluse.
- Bougrine, S., El-Majdouli, M.A., El-Imrani, A.A., 2017. A novel iterative improvement pivoting rule for local search heuristics, in: *GECCO'17: Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 13–14.
- Brimberg, J., Mladenović, N., Urošević, D., Ngai, E., 2009. Variable neighborhood search for the heaviest k -subgraph. *Computers & Operations Research* 36, 2885–2891.
- Brimberg, J., Salhi, S., 2019. A general framework for local search applied to the continuous p -median problem, in: Eisel, H., Marianov, V. (Eds.), *Contributions to Location Analysis: In Honor of Zvi Drezner's 75th Birthday*. Springer, Cham. volume 281 of *International Series in Operations Research & Management Science*, pp. 89–108.
- Cohen, J., 2013. *Statistical power analysis for the behavioral sciences*. Academic Press.
- Costa, L.R., Aloise, D., Mladenović, N., 2017. Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. *Information Sciences* 415-416, 247–253.
- da Costa, P., Rhuggenaath, J., Zhang, Y., Akcay, A., Kaymak, U., 2021. Learning 2-opt heuristics for routing problems via deep reinforcement learning. *SN Computer Science* 2, 1–16.
- Croes, G., 1958. A method for solving traveling-salesman problems. *Operations Research* 6, 791–812.
- Dawid, A., 2008. Entwicklung und Analyse eines Verfahrens zur effizienten Lösung des Tourenplanungs- und Laderaumoptimierungsproblems. Technical Report. Technische Universität Dortmund. Online reference at http://ls11-www.cs.tu-dortmund.de/_media/techreports/tr08-07.pdf, last access on March 17, 2024.
- Drugan, M.M., Thierens, D., 2012. Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies. *Journal of Heuristics* 18, 727–766.

- Du, J., Leung, J.Y.T., 1990. Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research* 15, 483–495.
- Erdelić, T., Carić, T., 2019. A survey on the electric vehicle routing problem: Variants and solution approaches. *Journal of Advanced Transportation* 2019. Online reference at <https://doi.org/10.1155/2019/5075671>, last access on March 17, 2024.
- Goos, P., Syafitri, U., Sartono, B., Vazquez, A., 2020. A nonlinear multidimensional knapsack problem in the optimal design of mixture experiments. *European Journal of Operational Research* 281, 201–221.
- Hackl, T., 2018. Local search methods for the particle therapy patient scheduling problem. Ph.D. thesis. Technischen Universität Wien.
- Hansen, P., Mladenović, N., 2006. First vs. best improvement: An empirical study. *Discrete Applied Mathematics* 154, 802–817.
- Hoos, H., Stützle, T., 2005. *Stochastic local search: Foundations and applications*. Elsevier.
- Irnich, S., Funke, B., Grünert, T., 2006. Sequential search and its application to vehicle-routing problems. *Computers & Operations Research* 33, 2405–2429.
- Krentel, M.W., 1988. The complexity of optimization problems. *Journal of Computer and System Sciences* 36, 490–509.
- Laporte, G., 1992. The Traveling Salesman Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 231–247.
- Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., 1985. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley.
- Lin, S., 1965. Computer solutions of the traveling salesman problem. *Bell System Technical Journal* 44, 2245–2269.
- Lin, S., Kernighan, B., 1973. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research* 21, 498–516.
- Lopes, R.B., Ferreira, C., Santos, B.S., 2016. A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Computers & Operations Research* 70, 155–162.
- Mjirda, A., Todosijević, R., Hanafi, S., Hansen, P., Mladenović, N., 2016. Sequential variable neighborhood descent variants: an empirical study on the traveling salesman problem. *International Transactions in Operational Research* 24, 615–633.
- Mladenović, N., Alkandari, A., Pei, J., Todosijević, R., Pardalos, P.M., 2019. Less is more approach: basic variable neighborhood search for the obnoxious p -median problem. *International Transactions in Operational Research* 27, 480–493.
- Mladenović, N., Todosijević, R., Urošević, D., 2016. Less is more: Basic variable neighborhood search for minimum differential dispersion problem. *Information Sciences* 326, 160–171.
- Moine, R., 2024. BI vs FI revisited. Online reference at https://github.com/Rob174/FIBI_Recherche, last access on January 31, 2024.
- Nascimento Silva, J.C., Coelho, I.M., Souza, U.S., Ochi, L.S., Coelho, V.N., 2020. Finding the Maximum Multi Improvement on neighborhood exploration. *Optimization Letters* 16, 97–115.
- Papadimitriou, C., Steiglitz, K., 1982. *Combinatorial Optimization*. Prentice-Hall.
- Pereira, T., Aloise, D., Brimberg, J., Mladenović, N., 2018. Review of basic local searches for solving the minimum sum-of-squares clustering problem, in: Pardalos, P., Migdalas, A. (Eds.), *Open Problems in Optimization and Data Analysis*. Springer, Cham, pp. 249–270.
- Pereira, T.C., 2017. *Novas heurísticas para o agrupamento de dados pela soma mínima de distâncias quadráticas*. Master’s thesis. Universidade Federal do Rio Grande do Norte.
- Potts, C.N., Van Wassenhove, L.N., 1985. A branch and bound algorithm for the total weighted tardiness problem. *Operations research* 33, 363–377.
- Rajab, R.S., 2012. *Some Applications of Continuous Variable Neighbourhood Search Metaheuristic (Mathematical Modelling)*. Ph.D. thesis. Brunel University.
- Reinelt, G., 1991. TSPLIB — A traveling salesman problem library. *ORSA Journal Computing* 3, 376–384.
- Resende, M.G., Ribeiro, C.C., 2016. *Optimization by GRASP: Greedy Randomized Adaptive Search Procedures*. Springer.
- Rosenkrantz, D.J., Stearns, R.E., Lewis II, P.M., 1977. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing* 6, 563–581.

- Sánchez-Oro, J., Sevaux, M., Rossi, A., Martí, R., Duarte, A., 2017. Improving the performance of embedded systems with variable neighborhood search. *Applied Soft Computing* 53, 217–226.
- Tang, H., 2008. Efficient implementation of improvement procedures for vehicle routing with time-dependent travel times. *Transportation Research Record* 2089, 66–75.
- Tari, S., Basseur, M., Goëffon, A., 2021. Partial neighborhood local searches. *International Transactions in Operational Research* 29, 2761–2788.
- Turan, B., Minner, S., Hartl, R.F., 2017. A VNS approach to multi-location inventory redistribution with vehicle routing. *Computers & Operations Research* 78, 526–536.
- Wilcoxon, F., 1945. Individual comparisons by ranking methods. *Biometrics* 1:80–83 1, 80–83.
- Wood, R.M., 2011. Modelling activities at a neurological rehabilitation unit. Cardiff University, United Kingdom.
- Yagiura, M., Ibaraki, T., 2002. Local search, in: Pardalos, P., Resende, M. (Eds.), *Handbook of applied optimization*. Oxford University Press, pp. 104–123.
- Yassine, A.A., Chidiac, R.H., Osman, I.H., 2013. Simultaneous optimisation of products, processes, and people in development projects. *Journal of Engineering Design* 24, 272–292.