A k-center-based Two-Phase Constructive Heuristic for the Depot-Free Multiple Traveling Salesperson Problem

ISSN 1870-4069

José Alejandro Cornejo-Acosta[1,2], Jesús García-Díaz[2,3], Blanca Verónica Zuñiga-Núñez[1]

[1] Tecnológico Nacional de México/ITS de Purísima del Rincón, Mexico

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico

[3] Consejo Nacional de Humanidades, Ciencias y Tecnologías, Mexico

{alejandro.ca, blanca.zn}@purisima.tecnm.mx, jesus.garcia@conahcyt.mx

Abstract. The Depot-Free Multiple Traveling Salesperson Problem (DFmTSP) is a variant of the classic Multiple Traveling Salesperson Problem (mTSP). In general, the purpose of the DFmTSP is that m salespersons must visit all the vertices of a given input complete weighted graph $G = (V, E, w)$ by minimizing an objective function. It has many applications in network optimization, routing, and logistics. Its main difference from other similar routing problems is that depots are not considered. In this paper, we introduce a Two-Phase constructive heuristic that uses an algorithm for the capacitated vertex k-center problem (CVKCP) in the first phase. For the second phase, a state-of-the-art heuristic for the classic TSP is used. The performed empirical evaluation shows that the proposal is capable of finding feasible and good-quality solutions in comparison to elaborated metaheuristics of the literature. Even more important, the proposal outperforms a novel metaheuristic when a percentage of imbalance between the number of vertices in the salespersons' paths is considered. Besides, one of the main advantages of the proposal is that it can find solutions in practical running times, which may be an important feature in certain situations.

Keywords: Network optimization, routing, heuristics, k-center, mTSP.

1 Introduction

Multiple Traveling Salesperson Problems (mTSPs) are a family of $\mathcal{NP}$-hard combinatorial optimization routing problems that generalize the classical Traveling Salesperson Problem (TSP). The mTSPs receive a complete weighted graph $G = (V, E, w)$ and a positive integer m as input. The purpose is to look for m paths for the salespersons that visit all the vertices in $V(G)$ by minimizing an objective function associated with the costs of the edges $E(G)$ [8]. In general, mTSPs can be applied in many routing and scheduling contexts, such as submarine patrol routing, bus routing, supervisor allocation, and some variants of job scheduling problems. Among the most popular studied variants of mTSP, two categories stand out:

- mTSPs that consider depots as part of the problem [12]:
  - SmTSP: All the salespersons must start their paths from a single defined vertex known as "depot".
  - MmTSP: There are multiple defined depots.
- mTSPs that do not consider depots at all [8]. These are known in the literature as Depot-Free mTSPs (DFmTSP).

Besides, mTSPs can also be classified according to the nature of the paths of the salespersons:

- Closed paths mTSPs (CP-mTSPs).
- Open paths mTSPs (OP-mTSPs).

In this context, a path is said to be closed if each salesperson starts and finishes its path at the same vertex. Otherwise, the path is said to be open. Among these mTSPs variants, DFmTSPs have received less attention in the literature in comparison with others, although they are considered one of the most fundamental variants [8]. This work focuses on the CP-DFmTSP. Besides, additional constraints to the maximum number of vertices each salesperson can visit are also considered. These are known in the literature as bounding constraints [12].

The rest of the paper is organized as follows: Section 2 reviews the related work of mTSPs and emphasizes the work focused on the DFmTSPs. Section 3 describes the proposal, a Two-Phase heuristic that uses the Cluster-First Route-Second strategy. Section 4 presents the carried out computational experimentation and performs an analysis of the results. Also, it discusses the advantages and disadvantages of the proposal. Finally, Section 5 states the conclusions and future work.

2 Literature Review

In general, mTSPs have been tackled through various optimization techniques, such as integer programming, approximation algorithms, exact algorithms, and heuristics and metaheuristics proposals. The initial integer programs (IPs) were proposed between 1960 and 1976 [16, 22, 9]. Over time, these IPs have been extended and improved to include considerations to address more realistic scenarios. [12] presented bounding constraints for these IPs for the SmTSP and MmTSP cases. In the context of the mTSP, bounding constraints specify that each salesperson must visit a minimum and/or maximum number of vertices. In the literature, most of the IPs have been proposed for mTSPs that consider depots as part of the problem. Nevertheless, recent advances in the literature have presented integer programs that are capable of dealing with both: mTSPs that consider depots and mTSPs that do not, and a combination between them [8]. Other recent proposals include the study of polyhedral approaches and branch-and-cut algorithms [2].

Besides IPs, other alternatives have been explored to approach mTSPs. As previously mentioned, mTSPs are $\mathcal{NP}$-hard. For this reason, in the last years, many researchers have proposed heuristic and metaheuristic algorithms in order to try to solve relatively big instances in practical running times. Among all the heuristics and metaheuristics proposed for mTSPs, evolutionary computing and some of their variants stand out [3, 26, 13].

Some recent proposals for the mTSP consist of hybridizing genetic algorithms with other metaheuristic algorithms. Such is the case of [11], where a genetic algorithm and an ant algorithm are combined, and [24], where a genetic algorithm is integrated with an invasive weed algorithm (IWO). Further, variants of ant algorithms have been used for similar routing problems like the mTSP with capacity and time windows [23] and the multi-objective Green Vehicle Routing Problem [15]. Other approaches that have been explored for mTSPs, are Two-Phase heuristics, which, as the name suggests, are procedures composed of two algorithm stages [1]. Among these, two main strategies stand out.

- **Cluster-First Route-Second**. The first phase consists of clustering the vertices; then, the second phase determines a feasible route for each cluster.
- **Route-First Cluster-Second**. The first phase consists of generating a large route that visits all the vertices; then, the second phase partitions such route into smaller routes.

These Two-Phase approaches have been widely used in some works for mTSPs. For SmTSP, in [3], a Route-First Cluster-Second strategy is used, and then a GA with intra-route heuristics is used to improve the quality of the routes. Other proposals have used the Cluster-First Route-Second, such as [25], where a variation of the k-means algorithm is used at the clustering phase, then a GA is used to build a route within each cluster. In fact, most works have used variations of the k-means algorithm for the clustering phase for mTSPs [14, 20, 19]. An interesting point of [19] is that the authors used a parallel approach to improve the running times. Regarding MmTSP the situation is similar, variations of the k-means clustering have been used in [21, 17]. It is worth noting that, for the second phase, most authors have used GAs, ant-based algorithms, and hybridizations between them.

It is important to remark that, although there are many proposals for the mTSPs, just a few focuses on the specific variant of this paper (DFmTSP). In fact, in the literature on mTSPs, many works study the SmTSP, but they refer to it just as the mTSP. On the contrary, there are also a few papers where the DFmTSP is studied but referred to as the MmTSP. This is deeply clarified in [8]. This paper studies the DFmTSP by considering two key points: depots do not exist in the problem statement, and there are bounding constraints. As far as we know, a few papers approach this specific variant. Among the last heuristic/metaheuristic proposals that consider these specific constraints, Zhou et al. [26] proposes a Partheno Genetic Algorithm (PGA) that considers lower-bound constraints. Also, [11] proposes a metaheuristic combining an Ant Colony and a PGA. This algorithm is called AC-PGA and considers both lower-bound and upper-bound constraints. According to the presented experimentation in [11], AC-PGA outperforms other proposals in terms of finding better quality solutions.

Regarding the objective functions for the mTSPs, two popular objective functions were initially considered in [4]. The first one is called *minsum* mTSP, where the objective is to minimize the sum of the cost of the salespersons' paths. The second objective function is known as the *minmax* mTSP, which consists of minimizing the longest path among the salespersons. However, the first one has become the most popular in the literature.

3 A Two-Phase Constructive Heuristic

This section introduces a Two-Phase constructive heuristic for the DFmTSP with upper-bound constraint. The proposal is based on the capacitated vertex k-center problem (CVKCP).

Along with k-means, k-center problems are natural clustering methods. In particular, the capacitated version imposes load-balance by considering an upper bound on the number of *clients* each center can attend. The vertex k-center problem (VKCP) has been used in the literature to design efficient constructive heuristics for the DFmTSP [18]. Nevertheless, it can not be used for the DFmTSP with bounding constraints since the VKCP does not restrict the maximum number of vertices that can be assigned to each center. As far as we know, the capacitated version has not been used as a clustering strategy for mTSPs in the literature. Thus, this work explores the advantages of using the capacitated vertex k-center problem as a clustering technique for the DFmTSP with bounding constraints. One of the main advantages, is that the CVKCP can create clusters with a maximum number of assigned vertices to each center, this characteristic is useful for the DFmTSP to limit the number of assigned vertices in each path, which is equivalent to the upper-bound constraint for mTSPs. Algorithm 1 shows the pseudocode of the Two-Phase proposed heuristic. The notation for this algorithm is the following:

- m is the number of salespersons.
- $p = \{p_1, p_2, \cdots, p_m\}$ is a solution for the DFmTSP composed by m salespersons paths.
- p_i is a salesperson path (a sequence of vertices) of a solution p.
- k is the number of centers for the CVKCP.
- C is the set of centers $C \subseteq V(G)$.

- P_C is the assignment function $P_C : V(G) \setminus C \to C$.
- U is the maximum number of vertices each salesperson can visit (upper-bound constraint).
- P_{c_j} is the subset of the assignment that contains only tuples of the form (u, c_j).
- $dom\left(P_{c_j}\right)$ is the set of vertices assigned to be covered by center c_j.

Algorithm 1: Two-Phase Constructive Heuristic.

Input: A weighted graph $G = (V, E, w)$, and two positive integers m and U

Output: A set of salespersons tours $p = \{p_1, p_2, \cdots, p_m\}$

$p \leftarrow \emptyset$
$k \leftarrow m$
$(C, P_C) \leftarrow kCenterClustering(G, k, U)$
foreach $c_i \in C$ **do**
    $X \leftarrow dom\left(P_{c_i}\right) \cup \{c_i\}$
    $p_i \leftarrow TSPRouting(G[X])$
    $p \leftarrow p \cup \{p_i\}$
end
return p

3.1 Clustering Phase

As mentioned before, the k-center algorithms have been used for clustering purposes. Thus, in this section we propose the usage of a heuristic that has proven to be effective in approaching the CVKCP. The used algorithm is known in the literature as the One-Hop Farthest-First heuristic (OHFF) [7]. This heuristic is based on an exact algorithm for the CVCKP [6], and exploits a relationship between the CVKCP and other combinatorial optimization problem known as the Minimum Capacitated Dominating Set (MCDS). Besides, in [6] is stated the CVKCP can be solved through a series of MCDS problems. The formal relationship is described in the Theorem 1 whose detailed proof can be consulted at [6]. It is known that the CVKCP and the MCDS are both NP-hard. Thus, in a general overview, the OHFF tries to solve the CVKCP by greedily trying to solve MCDS subproblems through a binary search. One of the main features of the OHFF is that parallel computing can be used to improve the running times. Algorithm 2 shows the pseudocode of the OHFF.

Theorem 1. *The minimum capacitated dominating set (MCDS) over the bottleneck graph $G_{OPT} = (V, E_{OPT})$ is the optimal solution to the CVKCP over the original input graph $G = (V, E, w)$, where OPT is the value of the optimal solution to the latter problem.*

Algorithm 2: One-Hop Farthest-First (OHFF) for the CVKCP [7].

Input: A complete weighted graph $G = (V, E, w)$, two positive integers k and U, and a non-decreasing list of the m edge weights of G, i.e., $w(e_1), w(e_2), ..., w(e_m)$, where $w(e_i) \leq w(e_{i+1})$

Output: A set of vertices $C \subseteq V$, such that $|C| = k$, and an assignment $P_C : V \setminus C \to C$

$high \leftarrow m$
$low \leftarrow 1$
$(C, P_C) \leftarrow (\emptyset, \emptyset)$
while $low \leq high$ **do**
    $mid \leftarrow \lfloor (high + low)/2 \rfloor$
    $(C', P_{C'}) \leftarrow GreedyMCDS\,(G, k, w(e_{mid}), U)$
    if $r(C', P_{C'}) \leq r(C, P_C)$ **then**
        $(C, P_C) \leftarrow (C', P_{C'})$
    end
    if $r(C, P_C) \leq w(e_{mid})$ **then**
        $high \leftarrow mid - 1$
    else
        $low \leftarrow mid + 1$
    end
end
while $|C| < k$ **do**
    $v \leftarrow \arg\max \{d\,(u, P_C\,(u)) : u \in V \setminus C\}$
    $P_C \leftarrow P_C \setminus \{(v, P_C\,(v))\}$
    $C \leftarrow C \cup \{v\}$
end
foreach $c_i \in C$ **do**
    $X \leftarrow dom\,(P_{c_i}) \cup \{c_i\}$
    $c_j \leftarrow \arg\min \{\max\{d(u, v) : v \in X\} : u \in X\}$
    $P_C \leftarrow P_C \setminus P_{c_i}$
    $P_{c_j} \leftarrow \{(v, c_j) : X \setminus \{c_j\}\}$
    $P_C \leftarrow P_C \cup P_{c_j}$
end
return (C, P_C)

3.2 Routing Phase

In the routing literature, many algorithms have been proposed for the classical Traveling Salesperson Problem (TSP) [5]. Among these proposals, there are some exact algorithms that guarantee to find the optimal solution. However, since TSP is $\mathcal{NP}$-hard, such algorithms may have an important limitation. Other proposals are approximation algorithms that do not guarantee finding an optimal solution but a solution inside a ratio of the optimal one. Likewise, heuristics do not guarantee optimality but are fast and very effective in finding near-optimal solutions. Metaheuristics are more elaborated procedures that use exploration and exploitation components to escape from local optimals during search. Lin–Kernighan heuristic (LKH) is one of the best heuristics for the TSP [10]. It is a local search algorithm that improves an input tour (Hamilton cycle) by exploring its neighborhood. Every time a shorter tour is found, the process is repeated until no better tour can be found. For this specific heuristic, a neighborhood is defined by considering the number of edges that are in one tour but not the other. For the routing phase of our proposal, we use the Lin-Kernighan algorithm since it has proven to be effective for TSP instances with thousand of vertices. Besides, an efficient implementation is provided in `http://webhotel4.ruc.dk/~keld/research/LKH/`.

4 Computational Experimentation and Analysis

We performed an empirical evaluation of the proposal over some instances of the TSPLIB dataset. For comparison purposes, we implemented the AC-PGA metaheuristic [11], which is one of the best metaheuristics for this specific variant of the problem. For the experimentation, we used the m values in the set $\{5, 10\}$, and for the upper bound U we used the values in $\{\lceil n/m \rceil, \lceil n/m \times 1.1 \rceil\}$. We refer to the value $\lceil n/m \rceil$ as a 0% of imbalance whereas $\lceil n/m \times 1.1 \rceil$ as a 10% of imbalance. The algorithms were implemented in the C++ programming language. All the experiments were carried out on a platform with Intel Core i9-13900, 64 GB RAM, under an OS Ubuntu 22.04.4 LTS 64-bit with a GCC 11.4.0 compiler. The version of the LKH is 2.0.10. All datasets and the implementation of the Two-Phase constructive heuristic can be consulted in `https://gitlab.com/alex.ca/DFmTSP-TP`. The parameter setting of the LKH used in our proposal, and the configuration of the AC-PGA are shown in Tables 1 and 2.

Tables 3 and 4 show the obtained results for 0% and 10% of imbalance of the three tested algorithms, where OHFF+ is the OHFF heuristic but executed $|V(G)|$ times with a different initial chosen vertex. The objective function is minsum. For the **AC-PGA** column, f_{best} is the objective value of the best-found solution of 30 independent runnings, whereas f_μ is the average, σ is the standard deviation, and $t(s)$ is the average running time in seconds. For the **OHFF/LKH** column, f_{best} is the objective value of the best-found solution of performing 30 independent runnings of the OHFF for the CVKCP and then running the LKH in the routing phase, f_μ is the average of the 30 runnings, σ is the standard deviation, and $t(s)$ is the average of the sum of both running times, the clustering phase and the routing phase. Due to the nature of the OHFF algorithm, some solutions for the CVKCP may include clusters with only one vertex.

In these cases, a salesperson path can not be constructed. Then, such solutions were ignored. For the **OHFF+/LKH**, the columns are the same as the previous but with the difference that OHFF+ was used. Note that the standard deviation of the latter is always 0 because the OHFF+ algorithm is deterministic. From these tables, we observe that when the percentage of imbalance is 0% (Table 3), the AC-PGA was capable of finding some of the best solutions.

Table 1. Parameter setting of the LKH [10].

Parameter	Value
RUNS	1
TIME_LIMIT	0.1s
MOVE_TYPE	5
PATCHING_C	3
PATCHING_A	2

Table 2. Parameter setting of the AC-PGA metaheuristic [11].

Parameter	Value
Population size	100
AC-PGA iterations	100
ACO iterations	100
ρ	0.1
α	2
β	8
γ	0.5

Nevertheless, an important remark is that the running times of AC-PGA are much higher than the proposals that use the OHFF as the clustering phase. For the case where 10% of imbalance is allowed (Table 4), most of the best solutions were found by the proposals that use the OHFF and the LKH. Thus, we conclude that, at least for the tested instances, the proposed Two-Phase heuristic is capable of finding better solutions when imbalance is allowed among the paths. Furthermore, the running times of the Two-Phase heuristic are many orders of magnitude lower than those of the AC-PGA metaheuristic.

Fig. 1 shows the printed solutions by the tested algorithms over the instance kroA200 with a 10% of imbalance. From this figure, we can observe that the best-found solutions by the AC-PGA contain some overlaps between the paths of the salespersons. On the contrary, solutions computed by the Two-Phase heuristic proposals have fewer overlaps, and the paths are better refined due to the LKH.

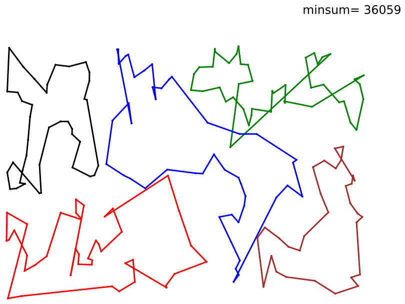
Fig. 2 and Fig. 3 show the convergence of the AC-PGA metaheuristic over the instance kroA200 with 0% and 10% of imbalance respectively. In these figures, the dotted lines represent the objective values of the solutions computed by the Two-Phase heuristic proposals. It is important to note that the Two-Phase heuristic proposals do not have generations since they are constructive heuristics. Nevertheless, they are shown in the figures for contrast purposes. In these figures, it can be observed that the proposals can find good quality solutions compared to the AC-PGA. However, due to the nature of AC-PGA metaheuristic, its exploration and exploitation components could cause the algorithm to escape from local optima, which could give the possibility that during the generations, the solution found may eventually be better than those found by the Two-Phase heuristic proposals, such is the case of the Fig. 2(b).

Table 3. Results over some instances of the TSPLIB dataset with 0% of imbalance. The best-found solutions are bold.

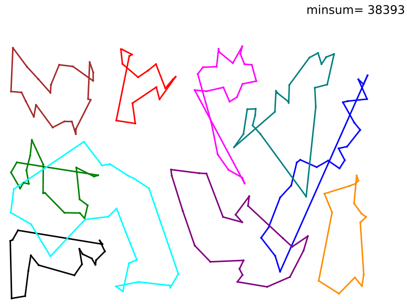
Instance	n	m	U	AC-PGA				OHFF/LKH				OHFF+/LKH			
				f_{best}	f_{μ}	σ	$t(s)$	f_{best}	f_{μ}	σ	$t(s)$	f_{best}	f_{μ}	σ	$t(s)$
kroA100	100	5	20	25016	25823	485.28	87	**23554**	27864	1640.57	0.0023	**23554**	23554	0	0.035
		10	10	**26593**	27371	487.34	86	29984	36203	2975.18	0.0033	30895	30895	0	0.058
kroB100	100	5	20	**24742**	26100	619.73	87	28223	30112	748.05	0.0022	24821	24821	0	0.036
		10	10	**27332**	28421	622.52	87	33756	40930	3799.44	0.0031	29802	29802	0	0.057
kroA150	150	5	30	30872	32510	618.19	199	**30513**	35621	2386.67	0.0027	31577	31577	0	0.098
		10	15	**33056**	34869	620.26	195	36779	44823	2763.87	0.0029	40862	40862	0	0.139
kroB150	150	5	30	30730	32320	565.49	199	29894	33882	3070.23	0.0023	**29224**	29224	0	0.095
		10	15	**33826**	35106	643.54	194	34829	46986	5240.63	0.0027	35186	35186	0	0.129
kroA200	200	5	40	35503	36994	722.84	355	34618	37381	1257.45	0.0020	**32284**	32284	0	0.206
		10	20	38968	40348	605.29	347	40126	46134	3567.72	0.0027	**35367**	35367	0	0.318
kroB200	200	5	40	34831	36393	577.19	355	**33191**	36795	1159.41	0.0022	34059	34059	0	0.218
		10	20	36222	38667	961.79	347	**35539**	47045	4400.87	0.0027	36715	36715	0	0.272
pr226	226	5	46	105262	107910	1586.42	452	**102824**	118669	9376.52	0.0016	107812	107812	0	0.167
		10	23	**110059**	114034	1961.42	442	126819	152624	17007.07	0.0026	125730	125730	0	0.233
pr264	264	5	53	**58210**	59999	895.17	617	60286	60864	443.44	0.0019	60552	60552	0	0.297
		10	27	53252	54709	831.13	608	**50868**	58201	7412.33	0.0024	51762	51762	0	0.301
pr299	299	5	60	58254	59787	687.26	791	55176	56996	2044.10	0.0022	**54629**	54629	0	0.351
		10	30	**62728**	64485	855.80	782	65223	72845	5111.33	0.0030	67588	67588	0	0.548
pr439	439	5	88	132677	136023	1630.25	1706	**117113**	121844	3746.26	0.0044	118061	118061	0	1.124
		10	44	**140475**	143925	1412.78	1682	140638	167431	13625.24	0.0076	145451	145451	0	1.803

Table 4. Results over some instances of the TSPLIB dataset with 10% of imbalance. The best-found solutions are bold.

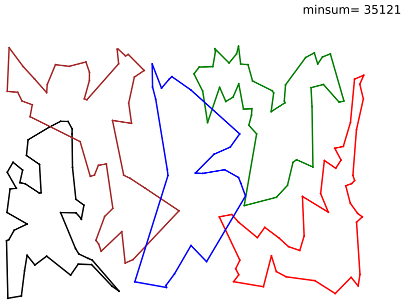
Instance	n	m	U	AC-PGA				OHFF/LKH				OHFF+/LKH			
				f_{best}	f_{μ}	σ	$t(s)$	f_{best}	f_{μ}	σ	$t(s)$	f_{best}	f_{μ}	σ	$t(s)$
kroA100	100	5	22	25116	25896	425.99	87	**24418**	28151	1706.43	0.0021	24499	24499	0	0.031
		10	11	26582	27463	413.00	87	**25605**	31282	3347.16	0.0032	26745	26745	0	0.051
kroB100	100	5	22	25651	26300	384.15	87	25651	29820	1790.54	0.0021	**25647**	25647	0	0.035
		10	11	**26262**	27985	901.65	87	27186	29427	1405.30	0.0029	27560	27560	0	0.052
kroA150	150	5	33	31301	32756	551.65	199	31393	34347	2015.50	0.0023	**29732**	29732	0	0.092
		10	17	33740	35252	667.75	196	32192	36397	3994.71	0.0026	**31820**	31820	0	0.110
kroB150	150	5	33	31251	32432	548.34	199	**28861**	32195	2963.20	0.0024	29601	29601	0	0.082
		10	17	32713	34371	646.82	196	**30496**	33561	1539.51	0.0031	31463	31463	0	0.111
kroA200	200	5	44	36059	37283	491.31	355	35121	38331	2317.41	0.0020	**32343**	32343	0	0.201
		10	22	38393	40014	609.98	348	36279	38180	2100.52	0.0027	**34599**	34599	0	0.246
kroB200	200	5	44	35456	36605	553.96	355	**33495**	37481	2284.02	0.0022	34716	34716	0	0.205
		10	22	37361	39150	655.16	349	**35087**	41324	4751.43	0.0028	35946	35946	0	0.244
pr226	226	5	50	**98788**	104878	2516.27	449	100707	111289	6742.86	0.0015	107997	107997	0	0.144
		10	25	**96528**	107709	3589.47	441	105342	129225	10807.50	0.0017	127776	127776	0	0.213
pr264	264	5	59	**59181**	61050	622.76	617	60789	60798	45.77	0.0020	60789	60789	0	0.263
		10	30	52405	54504	948.33	608	50480	55964	5127.32	0.0021	**50144**	50144	0	0.314
pr299	299	5	66	59125	60544	622.18	792	**52861**	58107	2706.32	0.0018	53936	53936	0	0.324
		10	33	62671	65695	1022.61	792	**56627**	62990	5636.68	0.0025	58514	58514	0	0.454
pr439	439	5	97	131110	135665	1700.13	1817	**115286**	119869	2922.18	0.0050	125807	125807	0	1.081
		10	49	139017	144026	1815.28	1682	**127688**	139649	6790.23	0.0042	137703	137703	0	1.413



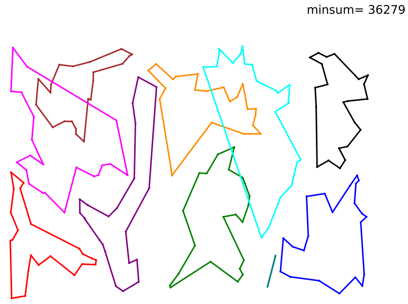
(a) AC-PGA, $m = 5$



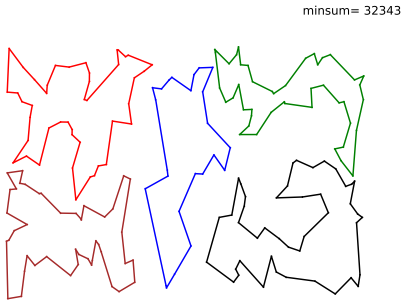
(b) AC-PGA, $m = 10$



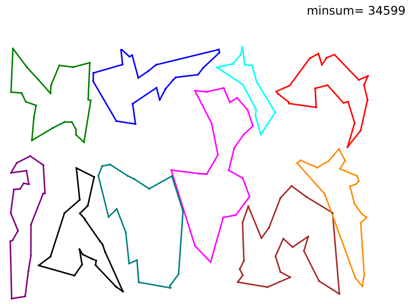
(c) OHFF/LKH, $m = 5$



(d) OHFF/LKH, $m = 10$

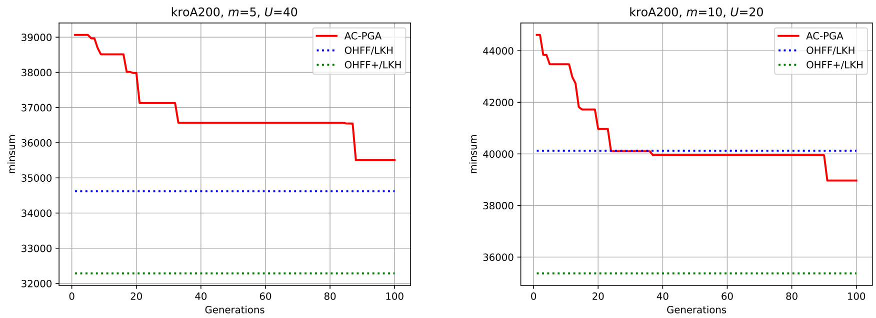


(e) OHFF+/LKH, $m = 5$



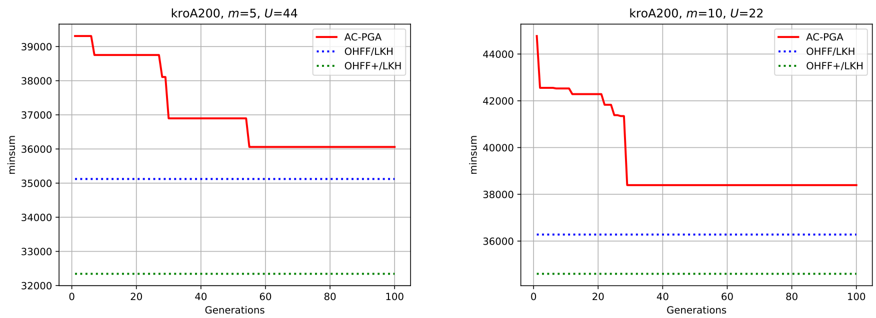
(f) OHFF+/LKH, $m = 10$

Fig. 1. Printed solutions of the kroA200 instance with 10% of imbalance.



(a) (b)

Fig. 2. Convergence plot of AC-PGA over instance kroA200 with 0% of imbalance. OHFF/LKH and OHFF+/LKH are included for comparative purposes.



(a) (b)

Fig. 3. Convergence plot of AC-PGA over instance kroA200 with 10% of imbalance. OHFF/LKH and OHFF+/LKH are included for comparative purposes.

5 Conclusions and Future Work

In this paper, a Two-Phase constructive heuristic for the Depot-Free Multiple Traveling Salesperson Problem (DFmTSP) was proposed. The main feature of our proposal is that a state-of-the-art heuristic for the capacitated vertex k-center problem (CVKCP) is used in the clustering phase, which is known as the One-Hop Farthest-First (OHFF). For the routing phase, a state-of-the-art heuristic called Lin-Kernighan (LKH) was used. The obtained results show that the proposed Two-Phase heuristic was able to find feasible and good-quality solutions in comparison with a state-of-the-art metaheuristic that employs elaborated exploration and exploitation mechanisms. Besides, reported running times support that the proposed Two-Phase heuristic is a good choice when practical running times matter. Some future work directions may arise from this research. For instance, working with dynamic clusters may be of interest. This can be performed through an evolutionary or metaheuristic that employs components to handle partitions of the set of vertices, then applying intensification methods over the partitions such as the LKH to find better quality solutions iteratively. This approach may lead to finding better solutions in comparison to working with static clusters/partitions, just as we worked in this research.

Acknowledgments. We acknowledge Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), and Tecnológico Nacional de México / ITS de Purísima del Rincón, for providing the necessary resources for the development of this research.

References

1. Anbuudayasankar, S. P., Ganesh, K., Mohapatra, S.: Models for Practical Routing Problems in Logistics. Springer Cham, 1 edn. (2014)
2. Barbato, M., Gouveia, L.: The Hamiltonian p-median problem: Polyhedral results and branch-and-cut algorithms. European Journal of Operational Research, vol. 316, no. 2, pp. 473–487 (2024) doi: 10.1016/j.ejor.2024.02.032
3. Bolaños, R. I., Toro O, E. M., Granada E, M.: A population-based algorithm for the multi travelling salesman problem. International Journal of Industrial Engineering Computations, vol. 7, no. 2, pp. 245–256 (2016) doi: 10.5267/j.ijiec.2015.10.005
4. Carter, A. E., Ragsdale, C. T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European Journal of Operational Research, vol. 175, no. 1, pp. 246–257 (2006) doi: 10.1016/j.ejor.2005.04.027
5. Cook, W. J.: In Pursuit of the Traveling Salesman. Princeton University Press, New Jersey, 1 edn. (2012)
6. Cornejo-Acosta, J. A., García-Díaz, J., Menchaca-Méndez, R., Menchaca-Méndez, R.: Solving the Capacitated Vertex K-Center Problem through the Minimum Capacitated Dominating Set Problem. Mathematics, vol. 8, no. 9, pp. 1551 (2020) doi: 10.3390/math8091551

7. Cornejo-Acosta, J. A., García-Díaz, J., Pérez-Sansalvador, J. C., Ríos-Mercado, R. Z., Pomares-Hernández, S. E.: A Constructive Heuristic for the Uniform Capacitated Vertex k-center Problem. ACM J. Exp. Algorithmics, vol. 28 (2023) doi: 10.1145/3604911
8. Cornejo-Acosta, J. A., García-Díaz, J., Pérez-Sansalvador, J. C., Segura, C.: Compact Integer Programs for Depot-Free Multiple Traveling Salesperson Problems. Mathematics, vol. 11, no. 13 (2023) doi: 10.3390/math11133014
9. Gavish, B.: A NOTE ON "THE FORMULATION OF THE M-SALESMAN TRAVELING SALESMAN PROBLEM". Management Science, vol. 22, no. 6, pp. 704–705 (1976) doi: 10.1287/mnsc.22.6.704
10. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. European Journal of Operational Research, vol. 126, no. 1, pp. 106–130 (2000) doi: 10.1016/S0377-2217(99)00284-2
11. Jiang, C., Wan, Z., Peng, Z.: A new efficient hybrid algorithm for large scale multiple traveling salesman problems. Expert Systems with Applications, vol. 139, pp. 112867 (2020) doi: 10.1016/j.eswa.2019.112867
12. Kara, I., Bektas, T.: Integer linear programming formulations of multiple salesman problems and its variations. European Journal of Operational Research, vol. 174, no. 3, pp. 1449–1458 (2006) doi: 10.1016/j.ejor.2005.03.008
13. Karabulut, K., Öztop, H., Kandiller, L., Tasgetiren, M. F.: Modeling and optimization of multiple traveling salesmen problems: An evolution strategy approach. Computers & Operations Research, vol. 129, pp. 105192 (2021) doi: 10.1016/j.cor.2020.105192
14. Latah, M.: Solving Multiple TSP Problem by K-Means and Crossover based Modified ACO Algorithm. International Journal of Engineering Research & Technology (IJERT), vol. 5, pp. 430–434 (2016)
15. Li, Y., Soleimani, H., Zohal, M.: An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. Journal of Cleaner Production, vol. 227, pp. 1161 – 1172 (2019) doi: 10.1016/j.jclepro.2019.03.185
16. Miller, C. E., Tucker, A. W., Zemlin, R. A.: Integer Programming Formulation of Traveling Salesman Problems. J. ACM, vol. 7, no. 4, pp. 326–329 (1960) doi: 10.1145/321043.321046
17. Mohammed, Y., Al-Khateeb, B.: A Novel Metaheuristic Algorithm for Multiple Traveling Salesman Problem. Journal of Advanced Research in Dynamical and Control Systems, vol. 10, pp. 2113–2122 (2018)
18. Na, B.: HEURISTIC APPROACH FOR THE NO-DEPOT K-TRAVELING SALESMEN PROBLEM WITH A MINMAX OBJECTIVE. Master's thesis, Texas A&M University (2006), https://hdl.handle.net/1969.1/5825
19. Othman, A., Haimoudi, E. k., Mouhssine, R., Ezziyyani, M.: An Effective Parallel Approach to Solve Multiple Traveling Salesmen Problem. In: Ezziyyani, M. (ed) Advanced Intelligent Systems for Sustainable Development (AI2SD'2018). pp. 647–664. Springer International Publishing, Cham (2019) doi: 10.1007/978-3-030-11928-7_58
20. Sathya, N., Muthukumaravel, A.: Two Phase Hybrid AI-Heuristics for Multiple Travelling Salesman Problem. International Journal of Applied Engineering Research, vol. 12, pp. 12659–12664 (2017)
21. Steven, A., Hertono, G. F., Handari, B. D.: Implementation of clustered ant colony optimization in solving fixed destination multiple depot multiple traveling salesman problem. In: 2017 1st International Conference on Informatics and Computational Sciences (ICICoS). pp. 137–140 (2017) doi: 10.1109/ICICOS.2017.8276351

22. Svestka, J. A., Huckfeldt, V. E.: COMPUTATIONAL EXPERIENCE WITH AN M-SALESMAN TRAVELING SALESMAN ALGORITHM. Management Science, vol. 19, no. 7, pp. 790–799 (1973) doi: 10.1287/mnsc.19.7.790
23. Wang, M., Ma, T., Li, G., Zhai, X., Qiao, S.: Ant Colony Optimization With an Improved Pheromone Model for Solving MTSP With Capacity and Time Window Constraint. IEEE Access, vol. 8, pp. 106872–106879 (2020) doi: 10.1109/ACCESS.2020.3000501
24. Wang, Z., Fang, X., Li, H., Jin, H.: An Improved Partheno-Genetic Algorithm With Reproduction Mechanism for the Multiple Traveling Salesperson Problem. IEEE Access, vol. 8, pp. 102607–102615 (2020) doi: 10.1109/ACCESS.2020.2998539
25. Xu, X., Yuan, H., Liptrott, M., Trovati, M.: Two phase heuristic algorithm for the multiple-travelling salesman problem. Soft Computing, vol. 22, no. 19, pp. 6567–6581 (2018) doi: 10.1007/s00500-017-2705-5
26. Zhou, H., Song, M., Pedrycz, W.: A comparative study of improved GA and PSO in solving multiple traveling salesmen problem. Applied Soft Computing, vol. 64, pp. 564–580 (2018) doi: 10.1016/j.asoc.2017.12.031