Proceedings Paper:
Copado-Méndez, P, Lin, Z, Barrena, E et al. (1 more author) (2022) K-Prototype Clustering Assisted Hybrid Heuristic Approach for Train Unit Scheduling. In: Communications in Computer and Information Science. EDCC22: Dependable Computing – EDCC 2022 Workshops, 12-15 Sep 2022, Zaragoza, Spain. Springer Nature , pp. 114-125.

https://doi.org/10.1007/978-3-031-16245-9_9

A K-Prototype clustering assisted hybrid heuristic approach for train unit scheduling

Pedro J. Copado-Méndez[1], Zhiyuan Lin[2⋆], Eva Barrena[3], and Raymond S. K. Kwan[4]

[1] Universitat Oberta de Catalunya, Rambla del Poblenou, 156, 08018 Barcelona, Spain
pcopadom@uoc.edu
[2] Institute for Transport Studies, University of Leeds, Leeds, LS2 9JT, UK
z.lin@leeds.ac.uk
[3] Pablo de Olavide University, Ctra. de Utrera, km. 1, 41013, Seville, Spain
ebarrena@upo.es
[4] School of Computing, University of Leeds, Leeds, LS2 9JT, UK
r.s.kwan@leeds.ac.uk

Abstract. This paper presents a K-Prototype assisted hybrid heuristic approach called SLIM+KP for solving large instances of the Train Unit Scheduling Optimization (TUSO) problem. TUSO is modelled as an Integer Multi-commodity Flow Problem (IMCFP) based on a Directed Acyclic Graph (DAG). When the problem size goes large, the exact solver is unable to solve it in reasonable time. Our method uses hybrid heuristics by iteratively solving reduced instances of the original problem where only a subset of the arcs in the DAG are heuristically chosen to be optimised by the same exact solver. K-Prototype is a clustering method for partitioning. It is an improvement of K-Means and K-Modes to handle clustering with the mixed data types. Our approach is designed such that the arcs of the DAG are clustered by K-prototype and each time only a small fraction of the arcs are selected to form the reduced instances. The capabilities of this framework were tested by real-world cases from UK train operating companies and compared with the results from running an exact integer solver. Preliminary results indicate the the proposed methodology achieves the same optimal solutions as the exact solver for small instances but within shorter time, and yields good solutions for instances that were intractable for the exact solver.

Keywords: Train Unit Scheduling · Hybrid Heuristics · Clustering · K-Prototype

1 Introduction

Many studies on passenger rolling stock scheduling in recent years have focused on Multiple Train Units (TU) which are the most commonly used passenger

⋆ Corresponding author

rolling stock in Europe and many other countries, because of their well-known advantages over traditional locomotives/wagons such as formation flexibility, energy efficiency, acceleration and shorter turnaround times. A TU is a reversible non-splittable fixed set of train cars, which can be coupled/decoupled with other units of the same or compatible types if it is needed. Nonetheless, railway operators still have to face high costs associated with leasing, operating and maintaining their fleets. Hence, an optimal schedule of these TUs reduces operation cost. Given a daily timetable of trips, a fleet of several types of TUs, routes, and station infrastructure: Train Unit Scheduling Optimization (TUSO) aims at determining an appropriate assignment plan such as each trip is covered by a single or coupled units in order to satisfy the passenger demand [31]. Note that TUSO is an NP-hard problem as is proved in [39, 7, 33].

One way for solving the TUSO problem in the UK's railways is a two-phase decomposition framework [29, 31], wherein the first phase the assignment of train units to trips is carried out ignoring some station layout details [31]. In the second phase, the fleet assignment is implemented taking account of station infrastructure to deal with shunting movements, unit permutation in a coupled formation and blockage of units, [28]. In this research we focused on the first phase. The aforementioned network flow level TUSO problem can be formulated as an Integer Multi-Commodity Flow Problem (IMCFP), which is based on a Directed Acyclic Graph (DAG) [7] representation, where the problem size is determined by the number of arcs. In [33], a branch-and-price method is designed to solve exactly small or medium-sized TUSO instances, but it is difficult to handle large instances due to its exact nature. In order to deal with this limitation, in [15] a hybrid heuristic approach called Size Limited Iterative Method (SLIM) is developed. In every iteration, the exact solver solves a reduced problems fast and comfortably, followed by an evaluation and modification of the arcs subset which is to be fed to the next iteration, such that the objective function will be no worse. Finally, the subset of arcs will converge to what is very close to an optimal solution. Local knowledge such as location, time, path of train connections in the DAG is used to decide which arcs are included in a reduced instance. For instance, arcs can be partitioned based on their time bands and in one particular iteration, all arcs from a band are included. This is referred to as "wheel rotation".

In this paper, we propose a novel approach for wheel rotation using K-prototype clustering. K-Prototypes is an upgraded version of K-Means [35] and K-Modes [21] suitable for mixed data types. It calculates the distance between numerical features using Euclidean distance (similar to K-means), but also calculates the distance between categorical features using the number of matching categories [22]. Based on their attributes, DAG arcs are thus grouped into clusters which are further used for creating reduced instances. This avoids explicitly using local knowledge to design strategies about which arcs to add in an iteration and thus the wheel rotation process is more generalised. Experiments based on real-world data from UK train operators show that the clustering-based SLIM

often outperforms the exact solver and can even successfully solve difficult large instances on which the exact solver will fail within reasonable time.

The remainder of this paper is organized as follows. A literature review is provided in Section 2, next we formally describe the problem in Section 3. Our solving approach is presented in Section 4. This is followed by an illustrative experimental evaluation provided in Section 5 and by conclusions in Section 6.

2 Literature Review

2.1 Rolling stock scheduling

There are several variants of problems studied involving rolling stock (train unit) planning in the literature. [39] first formulates the the rolling stock circulation problem (RSCP) as an integer multi-commodity flow problem on a single line with up to two train units that can be coupled. The objective is to minimize the number of units used. Issues such as train composition, attaching/detaching of units and unit blockage are not directly considered. A similar problem to [39] is studied by [4] and proposes an extended model where by introduce the concept of transition graph, unit orders in a coupled formation can be considered. [20] studies a variant RSCP where combining and splitting of trains are considered with a mixed integer programming model. [37] extends the problem scenarios of RSCP from a single line to multiple lines. Unit inventories are also described by extra decision variables. Branch-and-price is used to solve several real-world instances. The train unit assignment problem (TUAP) is first studied in [9], where it presents an integer multicommodity flow model. Since the maximum number of coupled units per trip is two, LP-relaxation can be enhanced in a precise manner with regard to the knapsack constraint [8]. Real-world instances of an Italian regional train operator with fleets of up to ten separate unit types and timetables containing 528-660 trains were solved. In [31], a two-phase strategy is presented for TUSO, in which the first phase allocates and sequences train trips to train units while temporarily disregarding station infrastructure specifics, while the second phase concentrates on completing the remaining station detail needs. In [33], a customised branch-and-price method for resolving the TUSO network flow level is given. Local convex hulls are utilised to enhance weak LP-relaxation bounds [32]. In [30], TUSO with bi-level capacity needs is investigated. In [28], station level unit conflicts are resolved by a feedback mechanism with added cuts. For larger and harder TUSO instances, a hybridized algorithm called size limited iterative method (SLIM) is developed in [16, 12]. It explicitly uses local knowledge such as location and time band to create small instances and is able to solve difficult instances failed by the exact solver.

2.2 Hybrid heuristics

The term of *Hybrid Heuristics*, or Hybrid Meta-heuristics [41, 11] normally refers to a class of algorithms for solving challenging combinatorial optimization problems. In a narrower sense, however, hybrid heuristics can be described as a

solution method in the following context: There is an exact solver available for solving the problem. However, due to the NP-hard nature of most combinatorial optimization problems, it can only solve small to medium instances. Therefore, an auxiliary heuristic method is developed to reduce the problem sizes in an iterative manner such that the small instances can be solved quickly and comfortably by the exact solver. In the early iterations, the solution quality from the reduced instances is often poor. However, the reduced instances are updated by customized search strategies in a way that in the final rounds of iterations most components needed for deriving optimal or near-optimal solutions will be included in the reduced instances.

A hybrid heuristic approach called PowerSolver is proposed in [26] for dealing with large and/or complex train driver scheduling problems. The relevant driver scheduling problems are solved by column generation [17] where a column represents a potential driver shift (duty). The corresponding set covering ILP model could have billions of columns, making the problem unsolvable. PowerSolver generates a series of tiny refined sub-problem instances that are fed into an existing efficient ILP-based solution. The usage of most relief opportunities (ROs, where/when a driver change can take place) is prohibited in problem instances, which reduces their sizes. A minimal collection of ROs is preserved in each iteration such that the following solution is no worse than the current best. A customized approach can help control the use of banned relief opportunities. It will also relax some of the restrictions placed on the problem instance before it is solved. PowerSolver provides a key step in fully automating the driver scheduling of UK train operators in large/complex real-world scenarios. It has been successful with many railway companies and has been routinely used as a key component of TrainTRACS, a commercial crew scheduling software suite [25]. A general hybrid metaheuristic method named Construct, Merge, Solve & Adapt (CMSA) is proposed by [6] for solving combinatorial optimization problems. CMSA generates a smaller sub-instance of the problem, which has a solution that is also feasible to its parent problem. A high quality solution is obtained by iteratively applying an exact solver to the reduced sub-instances. Strategies are designed such that feedback based on the results of the exact solver in a previous iteration will be provided to guide the parameter settings in next iteration. The effectiveness of CMSA was tested by two exemplar problems: the minimum common string partition problem and the minimum covering arborescence problem. From the experiment results, it is demonstrated that CMSA can achieve similar performance compared to the exact solver for small to medium sized instances, while its performance is significantly better than the exact solver when it came to large instances. See [3, 19, 5, 18] for recent research using CMSA.

2.3 K-Prototype clustering

K-Prototype was first proposed in [22], which combines K-Means (for numerical data) [35, 34] and K-Modes (for categorical data) [21]. The part for clustering categorical data was later improved by [10]. K-Prototype is able to cluster data of mixed types. Assume there are two mixed-type objects X and Y described by

their attributes $A_1^r, A_2^r, \ldots, A_p^r, A_{p+1}^c, \ldots, A_m^c$ (the fist p attributes are numerical and the remaining $m-p$ are categorical). Let x_j and y_j be the value of the j-th attribute of X and Y respectively. The distance between X and Y is measured by

$$d(X,Y) = \sum_{j=1}^{p}(x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j). \tag{1}$$

The first term is the (squared) Euclidean distance on numeric attributes while the second is a simple matching dissimilarity measure on the categorical attributes, i.e. $\delta(x_j, y_j) = 0$ if $x_j = y_j$ (same category) and $\delta(x_j, y_j) = 1$ otherwise. Weight γ is used to adjust the importance of the two types of attributes. Suppose the target number of clusters is given by $K > 0$, and there are n objects, the goal is to find the attribute values of each "centrtoid" of cluster l for attribute j, denoted by q_{lj} and whether an object i belongs to cluster l or not, indicated by binary variables $w_{il} \in \{0,1\}, 1 \le i \le n, 1 \le l \le K$ such that the total distance is minimised by [22].

$$\min_{w,q} \quad \sum_{l=1}^{K}\left(\sum_{i=1}^{n} w_{il} \sum_{j=1}^{p}(x_{ij} - q_{lj})^2 + \gamma \sum_{i=1}^{n} w_{il} \sum_{j=p+1}^{m} \delta(x_{ij}, q_{lj})\right) \tag{2}$$

subject to

$$\sum_{l=1}^{K} w_{il} = 1, \quad 1 \le i \le n. \tag{3}$$

The appropriate number of clusters K can be determined by the "elbow method" [24]. For the recent development of algorithms for clustering data with mixed type, see surveys [2, 38].

As a kind of machine learning algorithm for processing data, K-Prototype has been applied in various data science research, most of which are out of the scope of our paper. Apart from data science, there are a few cases where K-Prototype is used for applied problems. In [23], it assists optimising pavement lifecycle planning. [27] applies K-Prototype for the identification and analysis of vulnerable populations for malaria. In [40], K-Prototype has been applied in detecting anomaly intrusion activities. As far as the authors are aware, no application of K-Prototype has been used in improving optimisation algorithms for railway planing and management problems.

3 Problem Description

Train unit scheduling optimization (TUSO) concerns the assignment of train units to cover all the trips for an operational day, aiming at using the minimum number of train units while reducing the operational cost. It also allows the possibility of coupling and decoupling activities to achieve optimal use [31].

Our approach transforms the background problem TUSO into an Integer Multi-commodity Flow Problem (IMCFP) based on an initial DAG $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where the set $\mathcal{N}$ contains all the nodes (trips and sign-on/off nodes) and the set $\mathcal{A}$ contains all feasible arcs (feasible connections between nodes). Figure 1 illustrates a sample instance of five trips. More specifically, the node set $\mathcal{N} = \mathcal{N}_0 \bigcup \{s, s'\}$, where $\mathcal{N}_0$ represents the set of trips, and s and s' the *source* and *sink* nodes, respectively. Beside, each node is labelled with station origin/destination and also departure/arrival time. The arc set is defined as $\mathcal{A} = \mathcal{A}' \bigcup \mathcal{A}_0$, where $\mathcal{A}' = \{(i,j)|i,j \in \mathcal{N}\}$ is the *connection-arc* set and $\mathcal{A}_0 = \{(s,j)|j \in \mathcal{N}\} \cup \{(j,s')|j \in \mathcal{N}\}$ is the *sign-on/off* arc set. Each arc (i,j) stands for the potential linkage relation between trip i and trip j to be served by the same TU at the same station (same-location arc) or different stations (empty-running arc). Observe that the dashed arrow in Figure 1 represents an empty running arc.

Moreover, each arc a is labelled with the slack time a_t, which corresponds to the difference between the trip departure time of the successor node and the trip arrival time of the predecessor node, and cost a_c, which is defined by the waiting or empty running time between two trips linked by a. In the case of empty running arcs, the cost also considers the mileage. Finally, an $s - s'$ path in $\mathcal{G}$ represents a sequenced daily workload (the train nodes in the path in $\mathcal{G}$) for a possible unit schedule or diagram and the flow on it indicates the number of units used for serving those trains. The set of commodities K represents the set of TU types allowed. Note that a solution of IMCFP is a sub-graph of $\mathcal{G}$ such that all nodes are connected by paths from $s - s'$. Therefore, the optimal solution is the most *compact* sub-graph of $\mathcal{G}$, where compact refers to one minimum-cost subgraph. More details about the IMCFP and the DAG representation can be found in [33].

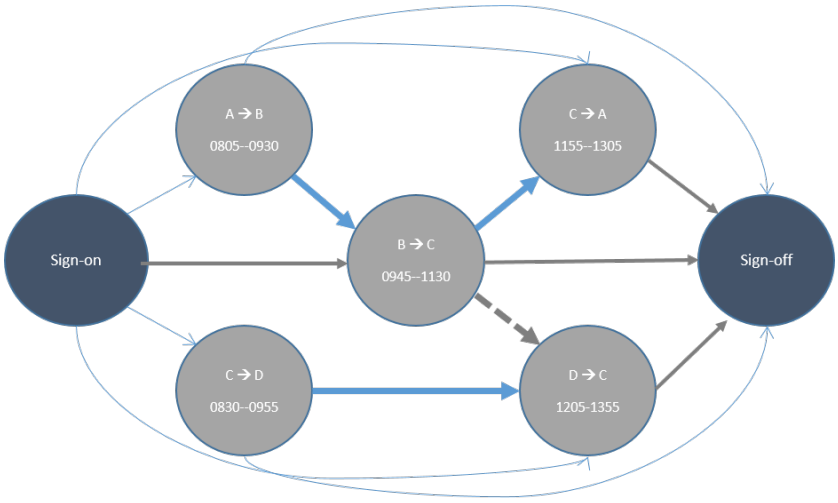

Fig. 1. Example of the initial DAG for an instance with five trips

4 Methodology

The proposed methodology is based on the hybridization technique introduced in [12, 13]. This methodology relies on the iterative resolution of sub-instances of the original graph instance $\mathcal{G}$. In each iteration, from a graph solution or *Essential Graph* $\overline{\mathcal{G}}$, which is characterised by containing an *Essential* number of arcs, an *Augmented Graph* $\widehat{\mathcal{G}}$ is constructed by extending the $\overline{\mathcal{G}}$ up to a fraction μ of $\mid A(\mathcal{G}) \mid$ in such way that $\mid \mathcal{A}(\overline{\mathcal{G}}) \mid < \mid \mathcal{A}(\widehat{\mathcal{G}}) \mid \ll \mid \mathcal{A}(\mathcal{G}) \mid$. Therefore, solving the problem on $\widehat{\mathcal{G}}$ yields a graph solution $\overline{\mathcal{G}}'$ which will be at least as good as $\overline{\mathcal{G}}$ and, after few iterations, it is expected to reach high-quality (sub-)optimal solutions in reasonable time.

The general SLIM+KP algorithm is described in the following and pseudo-coded in Algorithm 1. This requires the following input parameters: stopping criteria; l_{max}, that stands for the size of ranked solutions list; the aforementioned augmentation rate μ; and th, that is the number of sub-problems concurrently solving. The main loop works as follows: an initial feasible solution $\overline{\mathcal{G}}_0$ is inserted in the list of ranked solutions in lines 3-4 of Algorithm 1. The initial solution is constructed based on a first-in-first-out (FIFO) greedy heuristic [36, 20]. During the extraction phase of each iteration (lines 5-10), the algorithm randomly chooses an incumbent solution $\overline{\mathcal{G}}$ from the ranked list L. Then the algorithm produces $\widehat{\mathcal{G}}$ (augmentation phase), which is sent to exact method [33] to be solved. The exact solver yields a solution graph $\overline{\mathcal{G}}'$ and the algorithm iterates until one stopping criterion is satisfied.

Algorithm 1 SLIM+KP

Require: $\mathcal{G}$, l_{max}, μ, th
Ensure: $\mathcal{G}^*$
1: $L \leftarrow emptyList(l_{max})$
2: $Q \leftarrow emptyList(th)$
3: $\overline{\mathcal{G}}_0 \leftarrow initialSolution(\mathcal{G})$
4: $L \leftarrow insertSorted(\langle \overline{\mathcal{G}}^0 \rangle, L)$
5: **while not** $endCriteriaReached()$ **do**
6: $\quad \overline{\mathcal{G}} \leftarrow extraction(L)$
7: $\quad \widehat{\mathcal{G}} \leftarrow augmentation(\overline{\mathcal{G}}, \mu)$
8: $\quad Q \leftarrow sendToExactMethod(Q, \widehat{\mathcal{G}})$
9: $\quad$ **while** $th = \mid Q \mid$ **do**
10: $\quad\quad \langle \overline{\mathcal{G}}'_1, \ldots \rangle \leftarrow anyGraphSolutionReady?(Q)$
11: $\quad$ **end while**
12: $\quad L \leftarrow insertSorted(\langle \overline{\mathcal{G}}'_1, \ldots \rangle, L)$
13: **end while**
14: $\overline{\mathcal{G}}^* \leftarrow best(L)$

The heuristic controller holds the extraction and augmentation phases. In the extraction, the incumbent $\overline{\mathcal{G}}$ is chosen from the solution list L. This action

can be carried out using a solution randomly chosen overall graph solutions uniformly distributed in L. Regarding the augmentation phase, the augmented graph $\widehat{\mathcal{G}} = (\widehat{\mathcal{N}}, \widehat{\mathcal{A}})$ is built by setting the set $\widehat{\mathcal{N}}$ equal to set $\overline{\mathcal{N}}$ and the arc set $\widehat{\mathcal{A}} = \overline{\mathcal{A}} \cup \mathcal{H}$ is extended by means of the set $\mathcal{H} \subset \mathcal{A}$; as mentioned before, this set $\mid \mathcal{H} \mid$ is limited by the $\mu \cdot \mid \mathcal{A}(\mathcal{G}) \mid$. The set $\mathcal{H}$ is formed using a designed arc selection operator applied on arc clusters. This operator starts by selecting a random cluster in the first step, and collecting the arcs from this cluster as *circular* list in the second step. The construction of the clusters was carried out using K-Prototype methodology [22], which is carried out before executing the algorithm. As mentioned in Section 2.3, K-Prototype is a clustering method to deal with mixed data types. We have used the attributes contained in each arc to create the clusters. The description of these attributes is included in Table 1. Recall that each arc $v = (A, B)$ that links two trips A and B represents that the train units of trip A will also perform trip B.

Attribute	**Description**	**Type**
id_A	identifier of trip A	C
loc_A	location of arrival trip A	C
arr_A	arrival time of trip A to loc_A	N
ban_A	"true" if location loc_A is banned for coupling/decoupling operations, o/w "false"	C
id_B	identifier of trip B	C
loc_B	location of departure trip B	C
dep_B	departure time of trip B from loc_B	N
ban_B	"true" if location loc_B is banned for coupling/decoupling operations, o/w "false"	C
$slacktime$	difference between arr_A - dep_B	N

Table 1. Summary of the arc attributes used by K-Prototypes method for a given arc (A, B) between trips A and B. "C": categorical, "N": numerical.

5 Computational Experiments

To check the effectiveness of the proposed K-prototype assisted hybrid methodology, we have solved several instances from the data set described in [14], and we have compared our results with those obtained with the standalone exact method [33]. These instances differ in the number of nodes, arcs and fleet size. In order to solve each instance, we first create a partition of its arc set. For this purpose, we use the previously mentioned K-Prototype method [22]. In addition, to determine the optimal number of clusters, we have used the elbow method on the cost function provided by this method. Second, the instances are solved with the exact method within a maximum time of 2 hours. Finally, we repeat each experiment with SLIM, considering the previously generated clusters as heuristic

arc-selection operators, and setting the stopping criteria to the maximum time of 30 minutes. Regarding software, the exact method has been implemented in Mosel 3.0 and uses the Xpress MP 7.9 solver. SLIM has been developed in C#, while the K-Prototype method was included in the Python 3.6 module `kmodes` [1]. All the experiments have been performed on a computer with Windows Home 11, CPU Intel(R) i7-8750H and 16GB of RAM memory.

Results are shown in Table 2, which contains the following information for each instance. The first column refers to the instance name, which include the number of nodes and arcs (node#_arc#_type#). Second and third columns refer to the results obtained by the exact method and indicate the objective function value and the fleet size, respectively. Finally, the last three columns present the results obtained with the proposed K-prototype assisted hybrid methodology, indicating the number of clusters, the objective function, and the resulting fleet size. Observe that, for the cases where the exact solver achieves optimal solutions, the proposed hybrid methodology yields the same optimal objective values, and SLIM+KP is able to yield solutions for instances that are intractable for the exact solver.

	Exact Method		**SLIM +K-Prototype**		
Instance	**Obj**	**Fleet**	**#clusters**	**Obj**	**Fleet**
499_20151_2	-	-	4	85.747	85
510_2708_2	-	-	4	154.68	152
100_2164_1	19.412	18	4	19.412	18
358_3871_2	44.302	44	4	44.302	44

Table 2. Computational results from SLIM+K-Prototype and exact solver (exact solver ran for 120 min and SLIM+KP ran for 30 min)

6 Conclusions and future research

We have presented a K-Prototype assisted hybrid heuristic approach SLIM+KP to solve large instances of the Train Unit Scheduling Optimization problem. This problem can be modelled as an IMCFP based on a DAG, where each node represent a train trip and each arc (i, j) stands for the potential linkage relation between trip i and trip j to be served by the same train unit. For a fixed number of nodes, the higher the number of arcs, the higher the complexity of the problem. In order to solve large size instances, our hybrid methodology iteratively reduces the number arcs of the initial DAG and solve the problem based on the reduced graph by the exact method presented in [33]. The arc set is partitioned by means of the K-Prototype, a clustering method that can handle mixed data. This partition will then be the basis of the proposed K-Prototype assisted hybrid heuristic approach since arcs are selected depending on the cluster they belong to. By performing computational experiments on real-world cases from UK train

operating companies, we have compared our hybrid implementation against the exact solver [33]. The methodology has succeeded in achieving the same optimal solutions for small instances that could be solved exactly but with only about a quarter of time, and yields good solutions for instances that were intractable for the exact solver.

Future research includes more sophisticated strategies in SLIM+KP and to extend the applied cases to even larger real-world instances. We are also interested in further proposing an even more generalised approach for partitioning solution space (wheel rotation) for hybrid heuristics based on clustering and other methodological/algorithmic methods.

Acknowledgements We would like to thank Greater Anglia, Great Western Railway, Northern and Tracsis plc for supporting us with data for our research. This work was partly supported by grants PID2019-106205GB-I00, PID2019-104263RB-C41, and UPO-1263769. This support is gratefully acknowledged.

References

1. Kmodes, https://pypi.org/project/kmodes/
2. Ahmad, A., Khan, S.S.: Survey of state-of-the-art mixed data clustering algorithms. IEEE Access **7**, 31883–31902 (2019)
3. Akbay, M., Blum, C.: Application of cmsa to the minimum positive influence dominating set problem **339**, 17–26 (2021)
4. Alfieri, A., Groot, R., Kroon, L., Schrijver, A.:
5. Blum, C., Blesa, M.J.: A comprehensive comparison of metaheuristics for the repetition-free longest common subsequence problem. Journal of Heuristics **24**, 551–579 (2018)
6. Blum, C., Pinacho, P., López-Ibáñez, M., Lozano, J.A.: Construct, merge, solve & adapt a new general algorithm for combinatorial optimization. Computers & Operations Research **68**, 75–88 (2016)
7. Cacchiani, V., Caprara, A., Toth, P.: Solving a real-world train-unit assignment problem. Mathematical Programming **124**(1-2) (2010)
8. Cacchiani, V., Caprara, A., Maróti, G., Toth, P.: On integer polytopes with few nonzero vertices. Operations Research Letters **41**(1), 74–77 (2013)
9. Cacchiani, V., Caprara, A., Toth, P.: Scheduling extra freight trains on railway networks. Transportation Research Part B: Methodological **44**(2), 215–231 (2010)
10. Cao, F., Liang, J., Bai, L.: A new initialization method for categorical data clustering. Expert Syst. Appl. **36**, 10223–10228 (09 2009)
11. Christian Blum, G.R.: Hybrid Metaheuristics: Powerful Tools for Optimization. Springer Science & Business Media (2016)
12. Copado-Mendez, P., Lin, Z., Kwan, R.: Size limited iterative method: A hybridized heuristic for train unit scheduling optimization. CASPT 2018 (2017)
13. Copado-Mendez, P., Lin, Z., Kwan, R.: Size limited iterative method (slim) for train unit scheduling (2017)
14. Copado-Mendez, P.J., Lin, Z., Kwan, R.S.K.: Train units scheduling optimization (2018), http://archive.researchdata.leeds.ac.uk/id/eprint/537

15. Copado-Mendez, P.J., Lin, Z., Kwan, R.S.: Size Limited Iterative Method (SLIM) for Train Unit Scheduling. In: Proceedings of the 12th Metaheuristics International Conference, Barcelona, Spain.(2017). Leeds (2017)
16. Copado-Mendez, P., Lin, Z., Kwan, R.: Size limited iterative method (SLIM) for train unit scheduling. Proceedings of the 12th Metaheuristics International Conference, Barcelona, Spain (2017)
17. Desrosiers, J., Lubbecke, M.: A Primer in Column Generation, pp. 1–32 (03 2006)
18. Dupin, N., Talbi, E.G.: Matheuristics to optimize refueling and maintenance planning of nuclear power plants. Journal of Heuristics **27**, 63–105 (2021)
19. Ferrer, J., Chicano, F., Ortega-Toro, J.: Cmsa algorithm for solving the prioritized pairwise test data generation problem in software product lines. Journal of Heuristics **27**, 1–21 (04 2021)
20. Fioole, P.J., Kroon, L., Maróti, G., Schrijver, A.: A rolling stock circulation model for combining and splitting of passenger trains. European Journal of Operational Research **174**, 1281–1297 (2006)
21. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: In The First Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 21–34 (1997)
22. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery **2**(3), 283–304 (1998)
23. Karimzadeh, A., Sabeti, S., Shoghli, O.: Optimal clustering of pavement segments using k-prototype algorithm in a high-dimensional mixed feature space. Journal of Management in Engineering **37**(4), 04021022 (2021)
24. Koelbel, C.H., Loveman, D.B., Schreiber, R.S., Steele, G.L., Zosel, M.E.: Using MPI-2 PVM: Parallel Virtual Machine–A Users' Guide and Tutorial for Network Parallel Computing. Unstructured Scientific Computation on Scalable Multiprocessors Carolyn J. C. Schauble, and Gitta Domik (1991)
25. Kwan, R.: Case studies of successful train crew scheduling optimisation. J. Scheduling **14**, 423–434 (10 2011)
26. Kwan, R., Kwan, A.: Effective search space control for large and/or complex driver scheduling problems. Annals of Operations Research **155**, 417–435 (08 2007)
27. Li, C., Wu, X., Cheng, X., Fan, C., Li, Z., Fang, H., Shi, C.: Identification and analysis of vulnerable populations for malaria based on k-prototypes clustering. Environmental Research **176**, 108568 (07 2019)
28. Li, L., Kwan, R., Lin, Z., Pedro J Copado-Mendez, P.: Resolution of coupling order and station level constraints in train unit scheduling. Public Transport (2022)
29. Lin, Z., Kwan, R.S.K.: An integer fixed-charge multicommodity flow (FCMF) model for train unit scheduling. Electronic Notes in Discrete Mathematics **41**, 165–172 (2013)
30. Lin, Z., Barrena, E., Kwan, R.S.K.: Train unit scheduling guided by historic capacity provisions and passenger count surveys. Public Transport **9**(1-2), 137–154 (2017)
31. Lin, Z., Kwan, R.S.K.: A two-phase approach for real-world train unit scheduling. Public Transp **6**(1), 35–65 (2014)
32. Lin, Z., Kwan, R.S.K.: Local convex hulls for a special class of integer multicommodity flow problems. Computational Optimization and Applications **64**(3), 881–919 (2016)
33. Lin, Z., Kwan, R.S.: A branch-and-price approach for solving the train unit scheduling problem. Transportation Research Part B: Methodological **94**, 97–120 (2016)
34. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982)

35. Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: In 5-th Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)
36. Maroti, G., Gerards, A.M.H., Kroon, L.G., Eindhoven, T.: Operations research models for railway rolling stock planning (2006)
37. Peeters, M., Kroon, L.:
38. Preud'homme, G., Duarte, K., Dalleau, K., Lacomblez, C., Bresso, E., Smail, M., Couceiro, M., Devignes, M.D., Kobayashi, M., Huttin, O., Ferreira, J., Zannad, F., Rossignol, P., Girerd, N.: Head-to-head comparison of clustering methods for heterogeneous data: a simulation-driven benchmark. Scientific Reports **11** (02 2021)
39. Schrijver, A.: Minimum circulation of railway stock. CWI quarterly **6**, 205–217 (1993)
40. Srikanth, K., Reddy, S.R., Swathi, T.: A novel supervised machine learning algorithm for intrusion detection: K-prototype+id3. International Journal of Engineering Research Technology **3** (2014)
41. Talbi, E.G.: A taxonomy of hybrid metaheuristics. Journal of Heuristics **8**, 541–564 (01 2002)