A GRASP-based solution for real-time train route selection in disturbed railway traffic

Bianca Pascariu[1], Paola Pellegrini[1]

[1]Univertsité Gustave Eiffel, COSYS-ESTAS, F-59650 Villeneuve d'Ascq, France

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

Keywords: Train routing, Real-time rail traffic management, Greedy randomized adaptive search procedure

1 INTRODUCTION

Efficient rail traffic management is essential to ensure the timely operation of trains across rail networks. Typically, trains must follow a timetable planned well in advance. However, real-time operations are often subject to unpredictable disturbances. Dynamic adjustments of train timetables is thus required, to prevent congestion and minimize delays, addressed formally by the real-time Railway Traffic Management Problem (rtRTMP), a challenging, NP-hard problem.

The complexity of rtRTMP grows with the number of train routing options, expanding the solution space and computational difficulty. This motivates the formalization of the Train Routing Selection Problem (TRSP), a combinatorial problem focused on selecting the best set of routes for each train. The routes in the TRSP are selected based on cost estimates that align with the objective function of the rtRTMP, aiming to minimize the potential train delays that may arise from their use. These selected routes are then the only ones used in the subsequent rtRTMP solution, narrowing down route options. Despite its critical role in improving rtRTMP solution, the TRSP has received relatively limited attention in the literature.

Samà et al. (2016) formalized the TRSP as a minimum-cost clique problem, closely linked to clique-related problems. Due to the limitations of exact methods for large instances (Pascariu et al., 2021), heuristic and metaheuristic techniques were developed (Solnon & Bridge, 2006). This led Samà et al. (2016) to apply an Ant Colony Optimization (ACO) for the TRSP, later enhanced by Pascariu et al. (2022). ACO uses pheromone trails to reinforce successful route combinations, resulting in better performance in computation time and scalability. However, exploiting such a pheromone-based mechanism increases the computational burden, as maintaining and updating these trails requires computational resources. The usefulness of this exploitation for the TRSP has never been verified. To address this, we propose a Greedy Randomized Adaptive Search Procedure (GRASP) that explicitly explores the solution space without relying on pheromone-based learning. Research has shown that GRASP effectively solves various combinatorial optimization problems (Resende et al., 2016). Unlike ACO, GRASP employs a systematic exploration using greedy heuristics and randomization. This method avoids the complexity of pheromone management and offers a simpler, more efficient approach to solving the TRSP.

2 METHODOLOGY

Formally, the TRSP involves a set T of k trains, each requiring to traverse the railway network within a given time window. For each train $t \in T$, a set of alternative route assignments is available. We model the problem using the undirected k-partite graph $G = (V, E)$ proposed by Samà et al. (2016). Each vertex $v \in V$ corresponds to an alternative route for a train. The vertices are grouped into $k = |T|$ partitions such that for each train $t \in T$ we have an independent set of (all) alternative route assignments for t. Two vertices $v_i, v_j \in V$ are connected by an edge $e_{ij} \in E$ if they belong to different trains and represent compatible routes.

Constructing a clique of cardinality k in G corresponds to a feasible solution component of the TRSP, i.e., one route has been selected for each train and each pair of train routes is coherent. The optimization problem consists of finding the subset S_p of p minimum cost k-vertex cliques. Each vertex and edge in the graph has an associated cost. Hence, the total cost of a clique c is given by the sum of the cost of its vertices and edges, i.e., $\sum_{v_i \in c} u_i + \sum_{e_{ij} \in c} w_{ij}$.

In the TRSP, GRASP involves two main phases: a construction phase to build feasible cliques and a local search phase to refine them. GRASP repeats this iterative process until a computation time limit is reached, then selects the p best cliques as the final solution.

Clique construction. Starting with an empty clique, GRASP selects an initial vertex randomly to diversify solutions. Subsequently, GRASP uses a greedy-adaptive approach, where the cost for each candidate vertex considers both its route cost and the edge costs of its inclusion in the clique. To prevent convergence to local minima, GRASP introduces a probabilistic randomization mechanism: it selects the lowest-cost vertex with a probability of $1 - \sigma$ and randomly with probability σ. This phase terminates when the clique reaches the desired size k or when no feasible candidates remain. In cases where the clique is incomplete, GRASP uses a repair strategy.

Local Search. This phase improves each clique by replacing the vertex with the highest cost contribution with a more compatible candidate, reducing overall clique cost.

3 COMPUTATIONAL EXPERIMENTS

To evaluate the performance of the proposed GRASP algorithm for the TRSP, we conduct extensive computational experiments using real-world railway data from the Lille Flandres station area in France. We use 40 test instances based on a real-world timetable, each representing a one-hour window of perturbed traffic. The instances have an average of 39 trains per hour.

In the experiments, we assess the performances of solving the TRSP for the rtRTMP. Following the literature, we assign a computational time limit of 180 seconds for the solution of the overall TRSP and rtRTMP. Considering this overall limit, we perform experiments assessing different combinations of the computational time limits for the two problems. We solve the rtRTMP with the RECIFE-MILP solver proposed by Pellegrini et al. (2014) considering the minimization of the total exit delays as objective function. Overall, we compare the following solution approaches:

- no-TRSP: the rtRTMP is solved without the TRSP and considering all available routes in the rtRTMP. This is the classical solution setup of RECIFE-MILP (Pellegrini et al., 2014);
- pACO: the TRSP is solved with the parallel ACO algorithm proposed by Pascariu et al. (2022), along with the parameters that were defined in the same paper. The rtRTMP is then solved using exclusively the routes selected by the TRSP. This setup is the state-of-the-art regarding the solution of the TRSP;
- GRASP (current proposed method): the TRSP is solved with the GRASP we propose. The rtRTMP is then solved using exclusively the routes selected by the TRSP as in pACO. In the experiments, we test various values of the probability σ to select a random vertex during clique construction.

Following the literature (Pascariu et al., 2022) and as a result of preliminary analysis for the experimental setup, we select 5 cliques (maximum 5 routes per train) in the TRSP.

3.1 Results

Table 1 presents the results for all the experimental configurations tested, showing both the performance of the TRSP and rtRTMP. All performance data in the table represent the average across 40 instances per configuration. The first columns indicates the type of approach used for the TRSP and the configuration of the TRSP experiment: σ is the level of randomness in the GRASP-TRSP algorithm (0 for deterministic and higher values for increasing randomness), followed by the computation time (in seconds) dedicated to the TRSP solution. The next columns indicate the best and average clique costs of the top five cliques (in seconds). The last group of columns report data related to the rtRTMP: the computation time (in seconds) dedicated to the solution of the rtRTMP, the total secondary delay (in seconds), and the optimality gap (percentage difference from the optimal solution). When the rtRTMP solution relies on the TRSP pre-processing, the optimality gap accounts for the limited solution space including the subset of alternative routes, and thus it possibly refers to a local optimum for the overall instance. We use the relative optimality gap to determine if the solution is the best possible within the given route subset, or if further exploration could lead to better solutions.

The results in Table 1 show that both GRASP and pACO improve rtRTMP performance by selecting routes through TRSP. For pACO, longer TRSP times correlate with improved solutions due to more thorough exploration, with near-zero optimality gaps. For GRASP, the performance varies with the randomization parameter, σ.

At $\sigma = 0$, GRASP converges to higher-cost solutions, indicating limited exploration. Introducing a slight degree of randomness (σ = 5-10%) has a significant impact on the quality of the solutions. At this level, GRASP balances greedy selection with some exploration of less direct routes, improving both TRSP and rtRTMP performance. For example, at a 5-second TRSP time limit (σ = 5%), the best clique cost drops to 1323 seconds, an improvement of nearly 100 seconds compared to the purely greedy approach. This reduction in clique cost is reflected in the rtRTMP results, where the objective value improves to 818 seconds. As the TRSP time limit increases to 30 seconds, the best clique cost decreases further to 1258 seconds, and the rtRTMP objective value reaches 797 seconds. This is a significant improvement over pACO, which reaches an rtRTMP objective value of 867 seconds under the same conditions. These results suggest that controlled randomness enables GRASP to avoid local optima and perform well under short time limits. Higher randomization ($\sigma \geq$ 25%) leads to suboptimal results due to excessive exploration. Thus, moderate randomization is effective, particularly in time-constrained scenarios where GRASP consistently outperforms pACO.

An interesting observation is that some configurations of GRASP yield better results in the rtRTMP than the best obtained by pACO, despite having higher clique costs in the TRSP. For example, GRASP with a 5-second time limit and a σ = 5% has a higher clique cost than the best of pACO (1323 seconds compared to 1309 seconds), but it results in a lower rtRTMP objective value (818 seconds vs. 867 seconds). These results underscore that the TRSP, while contributing to rtRTMP performance, may not fully capture delay propagation, thus necessitating solutions that incorporate sufficient diversity to balance this inaccuracy and give flexibility to rtRTMP solution search. Regarding this point, GRASP performes still better than ACO. The results indicate that GRASP's controlled randomness allows it to explore a wider range of route combinations while maintaining a performance advantage by avoiding early convergence on suboptimal solutions. Instead, ACO focuses on a narrow selection of high-quality cliques, reinforced by pheromone trails that prioritize similar and reliable route combinations. Beyond the set of high-quality cliques in ACO, the quality of solutions begins to deteriorate.

In future research, we will focus on developing a TRSP algorithm that integrates a controlled level of diversity in TRSP solutions, able to enhance rtRTMP solutions. Moreover, we will

	TRSP				rtRTMP		
approach	σ (%)	time (s)	best cost (s)	avg cost (s)	time (s)	delay (s)	opt gap (%)
noTRSP	-	-	-	-	180	2395	85.8
ACO	-	5	1404	1429	175	937	0.2
	-	10	1344	1368	170	912	0.2
	-	20	1319	1339	160	876	0.2
	-	30	1309	1330	150	867	0.1
GRASP	0	5	1422	1521	175	852	0.6
	0	10	1422	1521	170	852	0.6
	0	20	1422	1521	160	853	0.6
	0	30	1422	1521	150	862	0.7
	5	5	1323	1391	175	818	1.4
	5	10	1308	1350	170	823	1.4
	5	20	1269	1317	160	812	1.3
	5	30	1258	1300	150	797	1.2
	10	5	1334	1431	175	853	1.9
	10	10	1314	1379	170	823	1.7
	10	20	1293	1354	160	784	1.4
	10	30	1285	1336	150	823	1.6
	25	5	1559	1661	175	916	3.3
	25	10	1494	1594	170	872	2.8
	25	20	1460	1536	160	887	3.3
	25	30	1443	1517	150	855	2.8
	50	5	2225	2407	175	1102	5.2
	50	10	2113	2275	170	1175	6.0
	50	20	2005	2175	160	1143	5.7
	50	30	1921	2109	150	1144	5.9
	75	5	2905	3230	175	1318	8.8
	75	10	2849	3103	170	1404	8.2
	75	20	2775	2958	160	1294	7.9
	75	30	2664	2876	150	1334	8.5

Table 1 – *Results of all experiment setups*

analyze solution evolution over time and evaluate the robustness of GRASP parameters across different problem settings.

References

Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D., 2016. Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B*, 85 (1), 89–108.

Pascariu, B., Samà, M., Pellegrini, P., D'Ariano, A., Pacciarelli, D., Rodriguez, J., 2021. Train routing selection problem: Ant colony optimization versus integer linear programming. IFAC-PapersOnLine, 54(2), 167-172.

Pascariu, B., Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., & Pacciarelli, D., 2022. Effective train routing selection for real-time traffic management: Improved model and ACO parallel computing. *Computers & Operations Research*, 105859.

Pellegrini, P., Marlière, G., Rodriguez, J., 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B*, 59 (1), 58–80.

Resende, M. G., & Ribeiro, C. C., 2016. Optimization by GRASP. Springer Science + Business Media New York.

Solnon, C., Bridge, D., 2006. An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems. In *System Engineering using Particle Swarm Optimization*. Nova Science publisher, 7–29.