Handling Vehicle Relocation Through Layered graphs

Alain Quilliot*[1,2] and Helene Toussaint[3]

[1]Laboratoire d'Informatique, de Modélisation et d'optimisation des Systèmes (LIMOS) – Université Blaise Pascal - Clermont-Ferrand II, Université d'Auvergne - Clermont-Ferrand I, CNRS : UMR6158 – Bât ISIMA Campus des Cézeaux BP 10025 63173 AUBIERE cedex, France

[2]University Clermont Auvergne (UCA) – LIMOS UMR CNRS/UCA618, Laboratoire Informatique, Modélisation et Optimisation des Systèmes, UCACNRS – Campus des Cézeaux, Clermont-Ferrand, 63000, France, France

[3]Centre National de la Recherche Scientifique, France (CNRS) – Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne, France – France

Abstract

A *one-way vehicle-sharing* system involves *stations*, together with free access *vehicles* (bicycles or electric cars), that users may pick up and give back at different *stations. Carriers* (trucks, drivers...) periodically move *vehicles* from *excess* stations to *deficit* ones. The *Vehicle-Sharing Relocation* (*VSR*) problem is about the design of the routes followed by *carriers* when relocating *vehicles.* We suppose here that *relocation* is performed by multi-task drivers concurrently to the system activity, as soon as some unbalanced situation is detected. We distinguish *time* and *cost* notions, impose some threshold to the *makespan* of the process, and consider *carrier-number* and *vehicle-riding-time* (time during which *vehicles* become unavailable) as part of the performance, together with the *carrier-riding-cost.*
We cast both *non-preemptive* and *preemptive VSR* (*carriers* may exchange *vehicles*) into the multi-commodity flow framework while using *layered graphs*, which extend *time-expanded networks*, and handle it according to a 3-step *vehicle-driven* approach: we first deal with a 1-layer projection and compute elementary connections followed by *vehicles* sharing same *carriers*; next we lift those connections into the layered graph through a multi-processor scheduling algorithm; finally we solve the restriction of our model to the resulting arc subset through a math-heuristic. We turn any *preemptive* solution into a *non-preemptive* one by solving an auxiliary *min-cost* flow model.

We perform experiments in order to compute lower bounds, evaluate heuristics and estimate the gap *preemption/non-preemption.*

Gavalas.D, al.: Design&management of vehicle-sharing systems: a survey, *ArXiv e-prints*, (2015).
Gouveia.L, al.: Layered graph approaches for combinatorial optimization problems; *C.O.R*, (2018).

Keywords: Routing, Scheduling, Layered Graphs

*Speaker