A multi-operator genetic algorithm for the diversified replenishment problem

Islem Kaabachi[1,2], Hiba Yahyaoui[1,2], Abdelkader Dekdouk[2], Saoussen Krichen[1]

[1] LARODEC, Institut Superieur de Gestion, University of Tunis, Tunisia
[2] College of Arts and Applied Sciences, Computer Science Department Dhofar University, Oman
{islamkaa@hotmail.com, hiba.yahyawi@gmail.com, d_abdulkader@du.edu.om, saoussen.Krichen@isg.rnu.tn}

Abstract. In this paper, we propose a Multi-operator Genetic Algorithm (MGA) for solving the diversified fuel replenishment problem (DFRP). We particularly consider the delivery problem of multiple types of fuels to *n* stations geographically dispersed in an addressed area. The main objective of the delivery process is to minimize the total travel distance of a fleet of trucks equipped by multi-compartment tanks. Numerous structural constraints are to be respected, namely assignment constraints that express the capacity limitations of each truck's compartment and routing constraints that state pathways' restrictions. The performance of the proposed algorithm is evaluated using a set of benchmark instances and compared to exact solutions generated by Cplex.

Fuel station replenishment, Vehicle routing problem, Multi-operator genetic algorithm.

1 Introduction

With the concern of running out of fossil fuel such as oil and gas, many countries are diversifying their fuel supply by considering renewable based fuels such as biofuels (ethanol, methanol, diesel and pyrolysis oil) and hydrogen fuel. This fuel supply diversification is an incentive to tackle the diversified fuel replenishment problem (DFRP), an extension of the petrol station replenishment problem (PSRP). This problem consists of optimizing the distribution of different types of fuel products from one storage point to a set of geographically dispersed stations. These products are transported in trucks equipped with multi-compartment homogeneous tanks. The objective is to minimize the driving distance of the tank trucks to all refueling stations. The DFRP problem investigated in this paper is a variant of the the vehicle routing problem with multi-compartment (VRP-MC). As the VRP-MC is *NP*-hard [9], several papers proposed meta-heuristic algorithms as the tabu search, memetic algorithm [11].
Cornillier et al. (2008a) [13] proposed an exact algorithm for the PSRP with a single period and a heterogeneous tank-trucks fleet. This algorithm was tested on randomly generated data and on a real-life case arising in Eastern Quebec. Taqa allah et al. (2000) [6] proposed two heuristics for solving multi-product petrol delivery with multi-compartment homogeneous trucks. The first heuristic deals with the case where the trucks can visit a station only once and the second one allows multiple visits for a station. It has been shown that for some benchmark instances that the first heuristic is less efficient than the second. Avella et al. (2004) [7] presents a real case of a company that distributes using heterogeneous tank trucks fleet, a set of fuel pumps located in an urban area with different types of fuel. Each customer places an order with a frequency of one or more days and each order must be satisfied in the following day. They described two solution approaches. The first one solves a large instances based on a routing heuristic and the second uses the Branch-and-Price approach to solve the problem. Ng et al. (2008) [12] studied two small petrol distribution networks in Hong Kong. They proposed a mathematical model that uses a simultaneous assignment of tank trucks with fuel stations, assuming that stations' inventories are managed by the vendor.
Our main contribution in this paper is twofold consisting in the writing of the mathematical formulation of the DFRP and the development of a new multi-operator genetic algorithm to solve the DFRP. We also used CPLEX to solve at optimality the generated instances.The proposed algorithm is indeed tested and compared on benchmarks from the literature.
The remainder of this paper is organized as follows: In section 2, we present the formal description of the problem and describe the mathematical formulation of the DFRP. In section 3, we present the new

genetic algorithm for solving the problem and its components are introduced and discussed. Section 4 reports computational results on a new sets of benchmark instances. Finally, some conclusions are drawn in Section 5.

2 Problem description

The DFRP can be modeled as a complete undirected graph $G = (V,E)$, where $V = \{0,1,\ldots,n\}$ is the vertex set, the depot corresponds to the node "0", and $E = \{(i, j)|i, j \in V, i \neq j\}$ is the edge set. A distance d_{ij} is associated to each edge $\{i, j\} \in E$. $V_c = V \backslash \{0\}$ is the set of station vertices.
A set P of fuel products distribution to stations that must be transported on independent compartments. Each tank-truck $k \in K$ has a dedicated compartment of fixed capacity Q_c for each product $p \in P$. Stations are served using a fleet of K homogeneous tank-trucks with p compartments and note that each truck $k \in K$ is limited by the capacity Q_{max}, thus the number of stations visited by each truck during the same trip cannot exceed the number of its compartments. For each product p, the station i requests an independent demand that is assumed to be fixed and equal to $q_{ip} \leq Q_c$.

3 Mathematical formulation

Let's introduce the notation of indices, parameters and decision variables to be used in the mathematical formulation $(1)-(14)$.

Indices	
$i,\ j,\ h$	Station index
k	Tank-truck index
p	Fuel product index

Parameters	
n	Total number of stations
K	Set of tank-trucks
d_{ij}	Travel distance between stations i and j
Q_{max}	Capacity of each truck k
q_{ip}	Demand of station i for each product p (Assume that $q_{0p} = 0$)
Q_c	Capacity of each compartment

Decision variables	
x_{ijk}	Binary variable that takes 1 if truck k travels from station i to j using truck k.
z_{jkp}	Binary variable that takes 1 if and only if product p of station j is loaded in truck k.
y_{ik}	Binary variable that takes 1 if and only if truck k services station i.
u_{ik}	Integer variable for every $i \in V_c$ representing the load in truck k after visiting station i.

The above parameter entries and decision variables of the DFRP give rise to the following mathematical model:

$$Min\ z(x) = \sum_{i \in V} \sum_{j \in V, j \neq i} \sum_{k \in K} d_{ij}\ x_{ijk} \qquad (1)$$

S.t.

$\sum_{k\in K} y_{ik} = 1$	$\forall i \in V_c$	(2)
$\sum_{k\in K} y_{0k} \leq \lvert K\rvert$		(3)
$\sum_{i\in V, i\neq j} x_{ijk} = y_{jk}$	$\forall j \in V_c, k \in K$	(4)
$\sum_{j\in V, i\neq j} x_{ijk} = y_{ik}$	$\forall i \in V_c, k \in K$	(5)
$\sum_{i\in V, p\in P} q_{ip}\, y_{ik} \leq Q_{max}$	$\forall k \in K$	(6)
$u_{ik} - u_{jk} + Q_{max}\, x_{ijk} \leq Q_{max} - \sum_{p\in P} q_{jp}$	$\forall i, j \in V_c, i \neq j$	(7)
$\sum_{p\in P} q_{ip} \leq u_{ik} \leq Q_{max}$	$\forall i \in V_c, k \in K$	(8)
$z_{jkp} \leq \sum_{i\in V} x_{ijk}$	$\forall j \in V_c, k \in K, p \in P$	(9)
$\sum_{k\in K} z_{jkp} = 1$	$\forall j \in V_c, p \in P$	(10)
$\sum_{j\in V_c} y_{jkp}\, q_{jp} \leq Q_c$	$\forall k \in K, p \in P$	(11)
$x_{ijk} \in \{0,1\}$	$\forall i, j \in V, k \in K$	(12)
$z_{jkp} \in \{0,1\}$	$\forall j \in V_c, k \in K, p \in P, q_{jp} \neq 0$	(13)
$y_{ik} \in \{0,1\}$	$\forall i \in V, k \in K$	(14)

The objective function (1) minimizes the driving distance for all routes performed by the tank-trucks. The first two constraints, (2) and (3) impose that exactly one truck services every station and each subset of the trucks leave the depot. Constraints (4) and (5) force each truck to arrive and leave from station only if it serves that station. Constraints (6) impose that the total demand delivered by each truck to stations is less than it capacity. Constraints (7) and (8) are subtour elimination constraints imposing both the capacity and connectivity of the feasible routes. Constraint (9) set z_{jkp} to zero for each product p if station j is not visited by truck $k \in K$. Constraints (10) mean that each product ordered by a station is brought by exactly one truck, while constraints (11) enforce the compartment capacity constraint for every truck. Finally, constraints (12), (13) and (14) state the binary nature of the decision variables.

4 Multi-operator genetic algorithm for the DFRP

This section is dedicated to introducing the Multi-operator genetic algorithm (MGA) that we propose to solve the DFRP. The general scheme of the meta-heuristic we propose is displayed in Algorithm 1.

The initial population is generated randomly across the search space. We start by describing the solution representation in section 4.1. A new population is generated from the current one through the selection, crossover, and mutation operators (Sections. 4.2, 4.3 and 4.4, respectively). The algorithm stops after a maximum number of iterations depending of the chromosome size.

Most of the studies stated in the literature have used genetic algorithms that use only single operator for crossover and mutation to resolve the DFRP. Here in our work, we present a multi-operator genetic algorithm simultaneously that uses two crossover and two mutation operators.

Algorithm 1 A MGA for the DFRP

```
Randomly generate an initial population IntPop of SizePop individuals
repeat
   Evaluate the fitness for each individual in the population
   Create the mating pool of IntPop size using Deterministic sampling
   while the mating pool is not empty do
      Select two parents at random from the mating pool
      Apply crossover operators (PMX or OP) to produce two offspring
      Apply mutation operators (SIM or EM) to each offspring
      Delete both selected parents from the mating pool
   end while
   Generation replacement
until stopping criterion
Record the current best solution
```

4.1 Solution encoding

The first step in designing a genetic algorithm for a particular problem is to devise a suitable solution representation scheme. We propose to encode a chromosome as an array in which each sequence of stations between two '0' refers to the pathway of each truck $k \in K$. Such an encoding is represented as follows:

0	1	6	3	0	2	4	...	0	8	7	5	0

$\underbrace{\quad}_{Truck_1}$ $\underbrace{\quad}_{Truck_2}$ ... $\underbrace{\quad}_{Truck_{|K|}}$

4.2 Mating selection

In this scheme, the selection of a parent is proportional to its fitness value. As all generated solutions are feasible in the proposed MGA, there is no need to take into account a penalty measure. The fitness function is computed in term of the driving distance cost of each solution as follows:

$$Fitness\ (X) = \sum_{i \in V} \sum_{j \in V, j \neq i} d_{ij}\ \sum_{k \in K} x_{ijk}$$

The notations are the same as mentioned in the section 3. The fitness function is expressed as the summation of the travel distance for all routes performed by truck k between stations i and j of the solution X.
Thus, at each generation of the MGA a mating pool of *PopSize* individuals is formed using a Deterministic Sampling selection operator [4]. In this operator, the average fitness of individuals is calculated as follows: $Avg\ (f_j) = \frac{\sum_{j=1}^{PopSize} f_j}{PopSize}$. Then the fitness value associated to each individual i is divided by the average fitness, but only the integer part of this operation $Int(\frac{f_i}{Avg\ (f_j)})$ is stored, where *PopSize* is the number of individuals in the population. If the value is equal or higher than one, the individual is copied to the mating pool. Remaining free places in the mating pool is fulfilled with individuals with the greatest fraction.

4.3 Crossover operators

Two crossover operators are used: the partially matched crossover (PMX) [3] and the One-point crossover (OP) [5]. The operators have the same probability of being used. The PMX operator is described by the following set of instructions:

a) Randomly choose two parents and two random crossover sites are generated;

b) Cut two substrings of equal size on each parent at the same positions;

c) Exchange the two selected substrings to produce proto-child;

d) Determine the mapping relationship based on the selected substrings.

e) Legalize proto-child with the mapping relationship.

In one-point crossover, two chromosomes are randomly selected from the currently processing population. The crossover that truncates two chromosomes reconstitutes two new ones by the concatenation of different fragments.

4.4 Mutation operators

After applying the crossover, we may get better and better chromosomes through the mutation operators. Two different mutation operators are used. The first is the Exchange Mutation (EM) [1], which chooses two genes randomly in the chromosome and their positions are exchanged. The second is a Simple Inversion Mutation (SIM) [2] selects randomly two cut points in the chromosome, and it simply reverses the allele between these two cut points. A uniform random number *prob* is generated. If $prob > Pm$ then the mutation process is activated on a random chromosome by applying the operator SIM otherwise the small changes are made on the truck route using the EM into account that the operator SIM has twice the probability of being used than the latter.

5 Computational experiments

The above presented mathematical model was implemented in IBM ILOG CPLEX 12.1. For the MGA algorithm, Matlab has been used. All experiments were performed under a personal computer with $Intel^{®}$ $Core^{TM}$ i7-4610M CPU @ 3.00GHz 3.00GHz 16 GB RAM and Windows 8.1 pro, 64-bit operating system, x64-based processor.
Two sets of numerical experiments were performed in order to evaluate the two approaches. In the first set, we compare the ability of the proposed algorithm with the exact solution approach using Cplex against enriched benchmark inspired from the literature [8] is available on [14].
We assume that:

- Each **station** i disposes its demand d_{ip} for each product p.
- Each **truck** k is characterized by:
  - Q_{max}: Capacity limit.
  - p: Number of compartments (which is equal to the number of products).
  - Q_c: Capacity limit of each compartment.
  - d_{ij}: Travel distance between stations i and j.
- **Ratio** $L = \frac{Q_{max}}{\sum_{i \in V_c} \sum_{p \in P} d_{ip}} \times 100$, informs about the average quantity to be loaded in each truck for the delivery process. This ratio is inversely proportional to the number of tank-trucks.
- Average relative percentage deviation **(ARPD)**: Percentage between the best solution $z(x^*)$ obtained using Cplex solver and the best found solution $z(x)$ obtained by the proposed MGA.

$$ARPD\% = \frac{z(x)-z(x^*)}{z(x^*)} \times 100$$

- **CPU**: Average computing time needed to obtain the best solution.

We drive an experimental investigation of the DFRP with a number of stations $n \in \{5, 10, 15\}$, trucks' capacities $Q_{max} \in \{1800, 2100, 2400\}$ and compartments' capacities $Q_c \in \{600, 700, 800, 900, 1050\}$ giving rise to 36 problems. After a series of runs, we have fixed the MGA parameters as displayed in table 1: Table 2 reports the numerical results of the DFRP instances using CPLEX and the MGA, followed with their CPU times, and the ARPD that corresponds to distance between exact and approxiamte solutions.
A second set of instances is proposed and compared with El Fallahi's metaheuristic solutions. For these problems, the well-known instances for the VRP from [10] have been transformed by [11] into set of VRP with multi-compartment (VRP-MC) instances with two compartments with fixed capacities and two

Parameter	**Type**
Stopping criteria	Number of iterations
Population size	*PopSize* $\in [5, 200]$, depending of the chromosome size
Selection phase	Deterministic sampling
Crossover	0.7
Mutation	0.3
Number of iterations	Depending of the chromosome size

Table 1: MGA tuning

Inst.	*n*	*L*(%)	Q_{max}	Q_c	*p*	CPLEX		GA		ARPD(%)
						$z(x^*)$	CPU	$z(x)$	CPU	
C_{01}	5	89.374	1800	900	2	1473.651	01.67	1473.653	0.31	0.00
C_{02}		107.197		900	2	816.879	01.97	816.879	0.072	0.00
C_{03}		101.058	2100	1050	2	2398.340	01.71	2398.341	0.08	0.00
C_{04}		98.637		1050	2	723.651	01.73	723.651	0.08	0.00
C_{05}		105.960	2400	800	3	1473.651	01.72	1473.652	0.10	0.00
C_{06}		86.052		800	3	2646.180	01.66	2646.180	0.06	0.00
C_{07}		111.940		800	3	377.690	01.76	377.692	0.10	0.00
C_{08}		92.592		800	3	377.690	01.55	377.692	0.11	0.00
C_{09}		82.834	1800	600	3	1804.461	01.59	1804.463	0.12	0.00
C_{10}		84.309		600	3	534.106	01.59	534.460	0.12	0.06
C_{11}		90.439	2100	700	3	3529.744	01.57	3529.746	0.12	0.00
C_{12}		85.854		700	3	3529.744	01.32	3529.746	0.10	0.00
C_{13}	10	20.129	1800	900	2	1473.651	01.93	1473.662	0.20	0.00
C_{14}		20.147		900	2	816.879	01.84	816.882	0.18	0.00
C_{15}		23.333	2100	1050	2	2398.340	01.86	2398.452	0.23	0.00
C_{16}		23.333		1050	2	723.651	01.98	723.674	0.22	0.00
C_{17}		26.666	2400	800	3	6940.121	01.67	6940.224	0.24	0.00
C_{18}		25.740		800	3	6940.121	01.72	6940.243	0.26	0.00
C_{19}		26.666		800	3	2256.216	01.99	2256.327	0.24	0.00
C_{20}		26.666		800	3	2256.216	01.84	2256.317	0.25	0.00
C_{21}		22.222	1800	600	3	5956.877	01.66	5956.927	0.27	0.00
C_{22}		22.222		600	3	2953.747	01.77	2953.851	0.26	0.00
C_{23}		25.925	2100	700	3	7989.178	01.87	7989.212	0.24	0.00
C_{24}		25.925		700	3	7989.178	01.82	7989.182	0.27	0.00
C_{25}	15	20	1800	900	2	10028.662	01.95	10028.667	0.29	0.00
C_{26}		20.120		900	2	10028.662	01.45	10028.671	0.31	0.00
C_{27}		20.100	2100	1050	2	8554.257	02.29	8554.268	0.33	0.00
C_{28}		23.333		1050	2	8554.257	02.19	8554.263	0.34	0.00
C_{29}		26.666	2400	1200	2	12890.805	02.12	12890.826	0.35	0.00
C_{30}		26.666		1200	2	7766.836	01.84	7766.841	0.34	0.00
C_{31}		25.740		800	3	10028.662	01.89	10028.681	0.36	0.00
C_{32}		25.472		800	3	10028.662	02.14	10028.673	0.34	0.00
C_{33}		20	1800	600	3	7134.959	01.79	7134.971	0.37	0.00
C_{34}		20.120		600	3	7134.959	01.98	7134.973	0.35	0.00
C_{35}		22.522	2100	700	3	12476.248	02.09	12476.264	0.38	0.00
C_{36}		22.436		700	3	12476.248	02.27	12476.257	0.37	0.00

Table 2: Computational results for small-sized DFRP

Inst.	n	Q_{max}	Q_c	Elfallahi et al. (2008)				This paper		$Gap(\%)$	
				MA		TS		MGA			
				Cost	CPU	Cost	CPU	Cost	CPU		
*vrpnc*1	50	160	80	**524.6**	23.7	**524.6**	15.8	548.284	3.20	4.51	4.51
*vrpnc*2	75	140	70	**842.7**	55.1	851.8	21.5	879.026	4.56	4.31	3.19
*vrpnc*3	100	200	100	853.2	62.3	**835.2**	95.3	847.190	5.12	**-0.70**	1.40
*vrpnc*4	150	200	100	1070.9	207.1	**1055.1**	411.2	1079.730	5.05	0.82	2.33
*vrpnc*5	199	200	100	**1330.3**	410.3	1348.8	809.2	1345.200	5.41	1.12	**-0.26**
*vrpnc*7	75	140	70	**928.1**	45.9	937.1	54.1	990.923	5.23	6.76	5.74
*vrpnc*8	100	200	100	**892.2**	57.0	894.5	105.2	892.292	5.32	0.01	**-0.24**
*vrpnc*9	150	200	100	1199.86	157.1	**1184.2**	395.6	1199.872	5.84	0.00	1.35
*vrpnc*11	120	200	100	1044.65	126.5	1043.8	64.9	**1043.741**	5.73	**-0.08**	0.00
*vrpnc*12	100	200	100	**819.6**	76.7	822.0	34.9	819.652	5.62	0.00	**-0.28**
Average				792.175	101.808	791.425	167.308	964.591	5.10	1.70	1.77

Table 3: Comparison with El Fallahi et al. (2008)

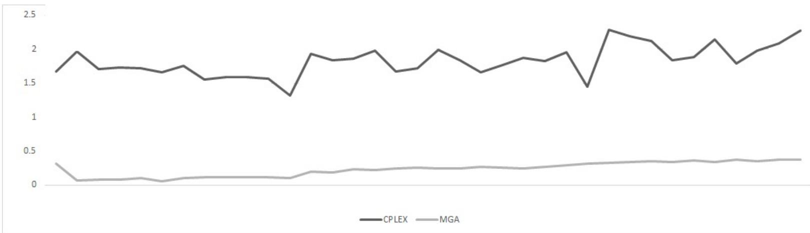

Figure 1: Run time of MGA and CPLEX

products by splitting the demand of each customer into two equal amounts. Table 3 states the results reported by El Fallahi et al. [11] that proposed two metaheuric approaches: a Memetic Algorithm (MA) and a Tabu Search (TS). Our results are compared to both approaches, in table 3, giving rise to two gaps.

$$Gap\% = Max(0, \frac{MGA-Metaheuristic}{Metaheuristic} \times 100)$$

Where Metaheuristic is set to TS and MA. The results are displayed in the last column of table 3, designed as a couple of gaps: (gap_1, gap_2). Note that the running times reported by El Fallahi et al. [11] are in seconds based on a PC Pentium 4 with 2.4 GHz.
Among the first set of experiments reported in table (2), we can notice that:

- The MGA is able to solve to optimality 99.99% of the instances within a CPU time close to 0.2 seconds.
- The CPU time between MGA and CPLEX is significantly important. Figure 1 plots such gap.
- The gap between state of the art approaches and MGA is slightly very small. We improved the solution cost for the instance *vrpnc*11 and reached the same solutions as the TS *vrpnc*8 and *vrpnc*12 while outperforming the MA.
- With respect to the whole testbed (Tables 2 and 3), the aveage gap is 0.85 which is very promising, given that the MGA requires a minimum runtime.

6 Conclusion

Despite the significant amount of studies related to the VRP, a limited list of these works take into consideration the multi-compartment option in the vehicles, that allows the transportation of heterogeneous products within the same vehicle. Throughout this paper, we have discussed the need of modelling the VRP with multi-compartment (VRP-MC) and presented the statement of the diversified fuel replenishment problem (DFRP) that is essential for several industries. We have developed the mathematical modelling of the DFRP as a VRP-MC specifically designed for the fuel framework. As the literature review revealed that such a modelling is *NP*-hard, we solved at optimality the DFRP using CPLEX. In addition, a new genetic algorithm is presented that uses multiple crossover and mutation operators to solve the DFRP is presented. Two sets of numerical experiments were performed in order to evaluate the two approaches. In the first set, the exact solution approach using Cplex and the multi-operator genetic algorithm (MGA) have been applied to relatively small. Our computational experiments showed that the proposed approach is able to solve to optimality 99.99% of the instances and got results for all generated instances. In a second set of experiments, the comparison with the algorithms by El Fallahi et al. (2008) [11] showed that our algorithm is competitive on their specific problem instances.

References

[1] Banzhaf, D. W. The "molecular" traveling salesman. Biological Cybernetics, 440 64(1), 7–14 (1990)
[2] Holland, J. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Abor (1975)
[3] Goldberg, D. E. and Lingle, R., Alleles, loci, and the TSP, in: Proc. 1st Int. Conf. on Genetic Algorithms, pp. 154–159 (1985)
[4] Goldberg, David E. Genetic Algorithms in Search, Optimization and Machine Learning. (1989)
[5] De Jong, K. An analysis of the behavior of a class of genetic adaptive systems, 447 (Ph. D. dissertation). USA: University of Michigan. (1975)
[6] Taqa allah, D., Renaud, J., Boctor, F.F. Le problème d'approvisionnement des stations d'essence. APII-JESA, Journal Européen des Systèmes Automatisés 34, 11–33. (2000)
[7] Avella, P., Boccia, M., Sforza, A. Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research 152, 170-179. (2004)
[8] Derigs,U., Gottlieb,J., Kalkoff,J., Piesche,M., Rothlauf,F., Vogel,U. Vehicle routing with compartments: Applications, modelling and heuristics. OR Spectrum 33, 885–914. (2011)
[9] Lenstra, J.K., Rinnooy Kan, A.H.G. Complexity of vehicle routing and scheduling problems. Networks. 11(2), 221–227 (1981)
[10] Christofides N, Mingozzi A, Toth P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C (eds) Combinatorial optimization. Wiley, Chichester, pp 315–338. (1979)
[11] El Fallahi A, Prins C, Wolfler Calvo R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Comput Oper Res 35(5), 1725–1741. (2008)
[12] Ng, W., Leung, S., Lam, J., Pan, S.. Petrol delivery tanker assignment and routing: a case study in Hong Kong. Journal of the Operational Research Society 59, 1191-1200. (2008)
[13] Cornillier, F., Boctor, F.F., Laporte, G., Renaud, J. An exact algorithm for the petrol station replenishment problem. Journal of the Operational Research Society 59, 607–615. (2008a)
[14] https://www.ac.tuwien.ac.at/research/problem-instances/