A New Simple and Effective Metaheuristic to solve the Vehicle Routing Problem with Cross Docking

Sanae LARIOUI, Rim MARAH
ENSIT Engineering School, Laboratory Systems, Control & Decision, Tangier, Morocco.
larioui@ensi.ma, Rim.marah@ensi.ma
Mohamed REGHIOUI, Abdellah EL FALLAHI
ENSATE, University of Abdelmalek Essaadi, Mhannech II, BP 2121 Tetouan Morocco

Abstract

This paper addresses a new version of the well-known vehicle routing problem with cross docking; we consider a more generalized version in which a customer orders different products from several suppliers. To solve this problem, we adapt a recently proposed mataheuristic based iterated local search. In particular, we focus on the feasibility tests for insertions and compare the new developed methods with those already existing. Finally, computational experiments on instances adapted from the classical vehicle routing problem with cross-docking are reported. Computational experiments with problems with 30 to 200 nodes indicate that the proposed algorithms are very competitive compared to previously proposed approaches like Memetic algorithm and tabu search.

Keywords

Iterated local search, Evolutionary local search, cross docking;

1. Introduction:

The cross-docking process, which can function as an efficient logistics strategy, includes three operations, namely receiving products from inbound vehicles, consolidating the products into groups according to their destinations, and shipping them on outbound vehicles. This process should be performed with minimum storage between operations. In fact, the cross dock replace the classical warehouse when the products transported do not need storage for a long time. Indeed, the cross dock transportation practice tries to eliminate the moves of material without added value as in the classical warehousing. In traditional warehouses, products are received and stored, generally for a significant time. This storage needs various operations which increase the cost of products without adding any value. The elimination of the non-added value moves ensures a significant reduction of the product's first cost and then increases the gain margin. In addition, cross docking reduces inventory investment, storage space, handling cost and order-cycle time, etc. However, it requires a very pernickety distribution management. This is why; its integration with the vehicle routing problem is very promising. However, the synchronization between the inbound and the outbound operations makes the Vehicle routing problem with cross docking much more complex than the classical VRP which is an NP-hard problem.

© IEOM Society International

The problem consists in defining a minimum cost set of routes for a fleet of vehicles that meets the demands of products for a set of suppliers and customers. The vehicles leave a single Cross-Dock (CD) towards the suppliers, pick up products and return to the CD, where products can be exchanged before being delivered to their customers. The vehicle routes must respect the vehicle capacity constraints, as well as the time window constraints.
In other words, the (VRPCD) can be defined as a set of known customer orders or requests, each one characterized by the cargo size, the pickup and delivery points. The orders are not driven directly from its origins to destinations, but they have to transit by the cross-dock terminal. The pickup and delivery processes are provided by a homogenous fleet of vehicles located at the cross dock. The objective is to minimize the total route cost respecting several operational constraints. In this paper we consider that the route cost is the total time of the pickup routes, the time necessary for the consolidation operations at the platform plus the delivery routes time.
Vehicle routing problems with cross docking are widespread in new logistics strategies but seldom studied in research literature. We address in this paper a generalization of the problem studied by Wen et al (2007), in which each customer is provided by only one supplier. Our version of the VRPCD considers that the demand of a given customer can be received from more than one supplier. Each node in the network representing the VRPCD must be visited by only one vehicle within its time window.

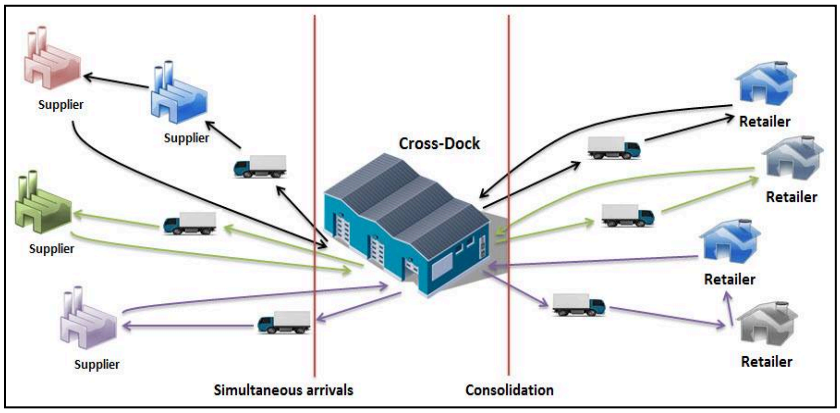

Figure 1: A cross docking distribution system

We note that taking into account the consolidation; the pickup and delivery routes are not independent. Trying to minimize the distance of the pickup and delivery routes separately is not sufficient; the exchanges of orders at the cross-dock also have to be taken into account.
The paper is organized as follows. The literature review is given in Section 2. Section 3 presents a summary of the mathematical model which has been covered in detail in Larioui et al.(2017).Section 4 describes the solution approaches proposed to solve the problem. The instances used and the obtained results are discussed in section 5. Finally, section 6 gives the conclusion and the perspectives of the work.

2. Literature review

Several research studies on cross-docking were conducted in the recent past. Most of the papers considered strategic issues such as the layout of the cross-dock (Bartholdi & Gue (2004)), and the best locations for such facilities (Gumus & Bookbinder (2004)). In the other hand, different metaheuristics like genetic, tabu search, particle swarm optimization and ant colony optimization has been developed to solve large scale optimization problems like the VRP and its different versions: VRPTW, Split Delivery Vehicle Routing Problem, Multi-Compartments Vehicle Routing Problem, etc. nevertheless, few papers have dealt with the Vehicle Routing Problem with Cross-Docking (VRPCD).
In particular, the problem consists in designing routes to pick up and deliver a set of transportation requests at minimal cost using a single cross-dock. It was introduced by Lee et al. (2006) in a variant which imposes trucks to arrive at the exact same time at the cross-dock. This constraint is too restrictive and cannot be considered generally, except in some few cases. Two years later, this constraint was relaxed by Wen et al. (2008), they consider only imposing precedence constraints based on the consolidation decisions and they also added time windows. This is the

most studied variant, and it is the one we will refer to in the paper. Authors have proposed several methods to solve it: based on tabu-search (Wen et al. (2007); Tarantilis (2012)), iterated local search (Morais et al. (2014)) and LNS (Grangier et al.(2017)). Many variants have been studied, with optional cross-dock returns (Santos et al. (2013)), with multiple cross-docks (Maknoon and Laporte (2017)).
In this paper, we consider a generalization of the one proposed in the literature. In fact, customer orders different products from several suppliers. The objective is to determine the optimal vehicle routing and scheduling in order to minimize the total time needed to supply customers. Since this problem is known to be NP-hard, three metaheuristics are developed to solve it. In the next section we present a reminder of the integer linear programming formulation for the VRPCD proposed in Larioui et al. (2017)

3. Mathematical formulation:

The VRPCD is defined on a network with a node set $N = \{P \cup O \cup D\}$. Where $P = \{1, \dots, n\}$ represents the set of pickup nodes, $D = \{1, \dots, m\}$ the set of delivery nodes and $O = \{o_1, o_2, o_3, o_4\}$ the set of cross docking nodes as presented above. Each node in N has a time window in which it could be visited, each customer $j \in \{D\}$ has a request equal to D_{ij}, from each supplier $i \in P$. Each node must be visited by only one vehicle within its time window.
E is the set of all the arcs of the network. They consist of arcs:
{(i,j): i,j ∈ P U { o1; o2},i≠ j}and the arcs {(i,j): i,j ∈ D U { o3; o4}, i≠ j}
K is the set of vehicles
The objective is to minimize the total transportation cost. The problem constraints can be classified into two categories: The first one concerns the vehicle routing, while the second category models the consolidation decisions at the cross-dock. The main interest of this model was to provide a compact problem specification: the problem is NP-hard, so only small instances could be solved by commercial MIP solvers. Cplex solver was not able to find optimal solutions for instances with more than 12 nodes. This is why metaheuristics are studied in the sequel to tackle the problem in practice. For more details about the model, the reader can consult Larioui et al. (2017).

4. Solution approaches

This section presents several methods to solve the VRPCD; these methods are compared with previously developed ones to find the best way to solve this type of problems. The approaches are the Iterated Local Search (ILS), and Evolutionary local search (ELS). The Best Insertion Heuristic (BIH) is used to build the initial solution for the proposed methods.

4.1 The best insertion heuristic:

This heuristic starts with an empty tour, and then, at each iteration it calculates the smallest insertion cost for each customer and performs the best insertion. When the remaining capacity is insufficient to accept new customers, a new tour is created. In our case, at each iteration of the BIH, the pickup node and its delivery nodes are inserted. The cost of the insertion is evaluated for all possible positions in all trips and the best location is selected for each node. The feasibility of insertion is evaluated respecting the vehicle capacity and time windows of each node. More details of this heuristic are given in Larioui et al. (2017). In the next section we will present a ILS and ELS to solve the problem. The choice of these methods was based on the great success of these methods in solving several problems in the literature

4.2 ILS and ELS for the vehicle routing problem with cross docking:

The ILS is a metaheuristic that involves the iterative application of a local search and the use of a perturbation as a diversification mechanism. It is a simple metaheuristic that needs only an initial solution, a local search and a perturbation mechanism. Its overall structure is described by figure 2. First, a constructive heuristic is used to generate a starting solution, which is converted to a local optimum by the local search. Then, the main loop of the algorithm executes ngen iterations or generations. At each iteration, a copy of the current best solution is changed randomly with the perturbation procedure (random move), acting as the mutation operator in the genetic algorithm. The resulting solution is then improved by local search and replaces the best solution only in case of improvement. The execution of the algorithm is stopped when a maximum number of iterations is reached.
Despite its simplicity, the ILS algorithm has proven to be a very successful approach to solve combinatorial optimization problems. For more information the reader can consult Stützle, T. (2006) and Vansteenwegen (2009). A detailed explanation of the ILS can be found in Lourenço et al. (2002). In the ILS developed in this paper, local search moves are the same as the ones used in tabu search presented in (larioui et al,.2017) . We notice that the splitting procedure is used for all methods. For more information about this procedure, the reader can consult our paper (Larioui et al.(2015).Figure 3 presents an overview of the ILS used

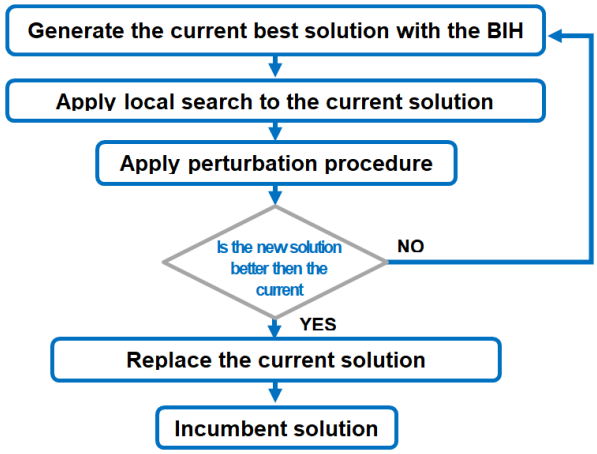

Figure 2: Flowchart of the ILS used

The Evolutionary local search (ELS) is an improved version of the ILS in which several solutions are derived from the current best solution instead of only one. It was introduced by Wolf and Merz (2007) .Starting with a solution built by the BIH and improved by a local search, the ELS perform ngen generations. Each generation build Nchild children-solutions. Each child is obtained from a copy of the current BestSol solution, which is then modified by the perturbation procedure and improved by local search. The best solution is saved in child-BestChildSol. If at the end of a generation the BestChildSol is better than the current solution BestSol, it is then replaced by BestChildSol for the next generation. The ELS recycles many components of the memetic algorithm developed in Larioui et al. (2015): The BIH (best-insertion heuristic), the giant tour, splitting procedure, and the local search moves.
We note that the perturbation mechanism is one of the two routines of ELS that modify the incumbent solution. It plays the role of a diversification tool in the general ELS framework since it performs one or several random moves on the current solution, based on customer relocation (Customers to relocate and insertion positions are picked randomly). Thus it can be interpreted as a mutation operator used in genetic algorithms.
The procedure is controlled with a parameter r denoting the number of operations to be performed. In our implementation, the operation is a removal and relocation of a node.

© IEOM Society International

It is noted that the perturb procedure is applied to the giant tour. It consists of relocating r nodes chosen randomly. Algorithm 1 presents the structure of the developed ELS.

Algorithm 1 –The ELS used

```
Initialize the random generator
Heuristic (BestSol)
LocalSearch (BestSol)
for gen := 1 to ngen do
    for child := 1 to nchild do
        ChildSol := BestSol
        Perturb (ChildSol)
        LocalSearch (ChildSol)
        UpdateSolution (ChildSol,BestChildSol)
    end for
    UpdateSolution (BestChildSol,BestSol)
end for
Return (BestSol)
```

5. Numerical experiments

We summarize in this section the numerical experiments performed with the proposed algorithms. To prove the performances of these approaches, the obtained results by the Memetic Algotrithm(MA),Tabu search algorithm(TS)(developed before), the iterated local search (ILS) and the evolutionary local search (ELS) were compared with those obtained by CPLEX for the small instances.

The instances are derived from the article of Wen et al (2007), with an adaptation to our problem. The test data consist of three Euclidean sets, each one containing 10 instances. Each instance has the same number of suppliers and customers known by their pickup and delivery locations (x; y). The time window for each pickup node is limited between 6:00 and 11:00 am. For delivery it is between 3:00 and 7:00 pm. The time horizon for the whole transportation operation is from 6:00 to 22:00. The demand transported from each pickup location to each delivery location is given in number of pallets. Vehicles drive at a constant speed of 60 km/h and have a capacity of 33 pallets. It takes ten minutes to prepare a vehicle, plus an additional one minute for each pallet to be loaded or unloaded.

To choose the best setting for the parameters a series of experiments has been realized to find the best tradeoff between the speed and the quality of solutions. The results of those experiments are reported bellow: The minimum cost spacing is Δ=0,2. Local search is applied with the probability pLS=0,1, vmax=100 indicates the maximum number of iterations. For ILS and ELS nchild=50 and ngen=50.

5.1 Computational results:

In this section, we report the different results of the methods developed, the first part concerns a comparison between the ILS and the ELS, the second concerns a convergence comparison, the last is for the results obtained by the branch and cut algorithm.

- Comparison between ILS and ELS.

As mentioned in section 3.3, the evolutionary local search (ELS) is an improved version of the iterated local search (ILS), due to the number of solutions extracted from the best solution; this method has been proved to be effective in providing high quality solutions. Table 1 shows a comparison between the results obtained with ILS and ELS. The first column is the file name; the second indicates the size of the problem, the third and fourth show the results obtained with the two methods. The last one gives the GAP between the two values for every instance.

© IEOM Society International

Table 1. Comparison between ILS and ELS results

FILE	CLIENT	ILS		ELS		
		Cost	Time(s)	Cost	Time(s)	Gap(%)
DATA10_A	20	1885,279	0,761	**1852,601**	0,74	1,76
DATA10_B	20	2261,668	0,8	**2213,163**	0,682	2,19
DATA10_C	20	1737,642	0,875	1737,642	0,449	0
DATA10_D	20	1898,498	0,7	**1878,91**	0,686	1,04
DATA30_A	60	5463,9	3,412	**5334,241**	2,805	2,43
DATA30_B	60	5792,84	3,158	**5575,543**	3,101	3,89
DATA30_C	60	6879,245	2,744	**6870,951**	3,262	0,12
DATA30_D	60	5475,121	3,614	**5462,222**	3,049	0,23
DATA50_A	100	8212,331	7,074	**7877,638**	6,988	4,24
DATA50_B	100	9743,713	7,659	**9665,1**	9,408	0,81
DATA50_C	100	9275,662	8,225	**9094,949**	5,341	1,98
DATA50_D	100	9777,274	8,565	**9768,53**	5,565	0,08
DATA50_E	100	9634,931	7,823	**9377,383**	7,183	2,74
DATA50_F	100	10216,785	6,098	**9991,617**	7,066	2,25
Average		**6303,92064**	**4,39342857**	**6192,89214**	**4,02321429**	1,69

On the basis of those results, it is clear that the ELS outperform the ILS for the majority of instances (more than 83%), with almost the same or better computational time, which proves the efficiency of deriving several solutions from the best instead of one.

- Results

Table 2 shows a comparison between BIH, ELS and TS. The first column contains the file name, the second one indicates the size of the problem (number of nodes),the results of the BIH is given in the third column, the next three columns show the results obtained with ILS, indicating the CPU and the GAP comparing with BIH. Then, the same comparison is given for ELS and for TS In the next columns

Table 2. A general comparison

FILE	CLIENT	BIH	MA		ILS		ELS		TS	
			cost	time	Cost	Time	Cost	Time	Cost	Time
DATA10_A	20	1958,336	1906,224	0,8	1885,279	0,761	1852,601	0,74	**1814,48**	1,25
DATA10_B	20	2607,953	2276,194	0,55	2261,668	0,8	2213,163	0,682	**2208,24**	0,89
DATA10_C	20	2084,266	1856,265	0,64	**1737,642**	0,875	**1737,642**	0,449	**1737,64**	1,52
DATA10_D	20	2239,171	2174,088	1,23	1898,498	0,7	1878,91	0,686	**1839,05**	1,15
DATA10_E	20	2177,27	2029,263	0,75	1882,679	0,812	1882,679	0,709	**1881,09**	1,31
DATA10_F	20	2207,283	**1790,135**	1,06	1984,656	0,764	1984,656	0,599	1923,58	1,05
DATA10_G	20	2177,916	2000,594	0,81	1929,268	0,724	1929,268	0,577	**1853,47**	1,16
DATA10_H	20	2120,837	1956,153	1,15	1952,141	0,58	1950,426	0,59	**1945,21**	1,15
DATA10_I	20	2504,422	**2075,5**	0,98	2204,55	0,764	2130,583	0,61	2116,72	0,89
DATA10_J	20	2348,227	1948,906	0,73	1949,744	0,56	1949,744	0,711	**1941,65**	1,04
DATA30_A	60	7934,878	7474,786	3,85	5463,9	3,412	**5334,241**	2,805	5628,37	13,24
DATA30_B	60	8097,81	7555,811	2,34	5792,84	3,158	**5575,543**	3,101	5849,16	11,38
DATA30_C	60	8412,235	**6148,789**	6,02	6879,245	2,744	6870,951	3,262	6804,38	9,16
DATA30_D	60	8138,668	7042,106	5,85	5475,121	3,614	**5462,222**	3,049	5547,99	11,72
DATA30_E	60	8202,176	5986,499	4,52	5843,622	2,923	**5710,351**	3,432	5849,25	10,00
DATA30_F	60	7777,353	7131,365	5,3	5349,624	3,032	**5272,579**	2,706	5563,75	12,65
DATA30_G	60	7694,181	6774,495	2,53	5322,045	3,143	**5003,553**	3,518	5343,09	10,84
DATA30_H	60	8594,752	7062,248	3,46	6350,341	2,65	**6081,202**	3,18	6340,95	11,96

© IEOM Society International

DATA30_I	60	7733,188	6863,495	3,99	5604,428	3,194	**5583,91**	3,246	5643,63	11,91
DATA30_J	60	8011,966	6303,393	6,11	5472,222	3,568	**5440,213**	3,893	5572,18	13,24
DATA50_A	100	12822,425	11096,202	7,8	8212,331	7,074	**7877,638**	6,988	8462,29	54,15
DATA50_B	100	14268,416	10796,803	11,58	9743,713	7,659	**9665,1**	9,408	10119,01	34,83
DATA50_C	100	13153,539	11281,22	9,81	9275,662	8,225	**9094,949**	5,341	9304,85	49,23
DATA50_D	100	13822,876	10831,557	7,82	9777,274	8,565	**9768,53**	5,565	10286,89	45,80
DATA50_E	100	13639,363	10445,996	10,34	9634,931	7,823	**9377,383**	7,183	9679,82	56,94
DATA50_F	100	14122,077	10998,258	10,47	10216,785	6,098	**9991,617**	7,066	10136,43	47,11
DATA50_G	100	13518,932	10675,237	9,55	8527,801	6,994	**8513,966**	6,701	8896,67	46,30
DATA50_H	100	13326,85	10574,697	9,97	8971,647	6,475	**8889,097**	5,148	9328,71	43,21
DATA50_I	100	13124,025	10129,632	9,89	9991,168	7,304	**9693,447**	6,368	10054,40	45,36
DATA50_J	100	13204,646	10008,955	9,97	8635,023	8,13	**8480,076**	6,402	8855,30	40,58
Average	**average**	**7934,201**	**6506,495533**	**5,00**	**5674,19493**	**3,77**	**5573,208**	**3,49**	**5750,94137**	**19,70**

- Convergence comparison:

The curves below show a comparison between the results of the ELS and TS, the first represents the cost, while the second is related to the computational time.

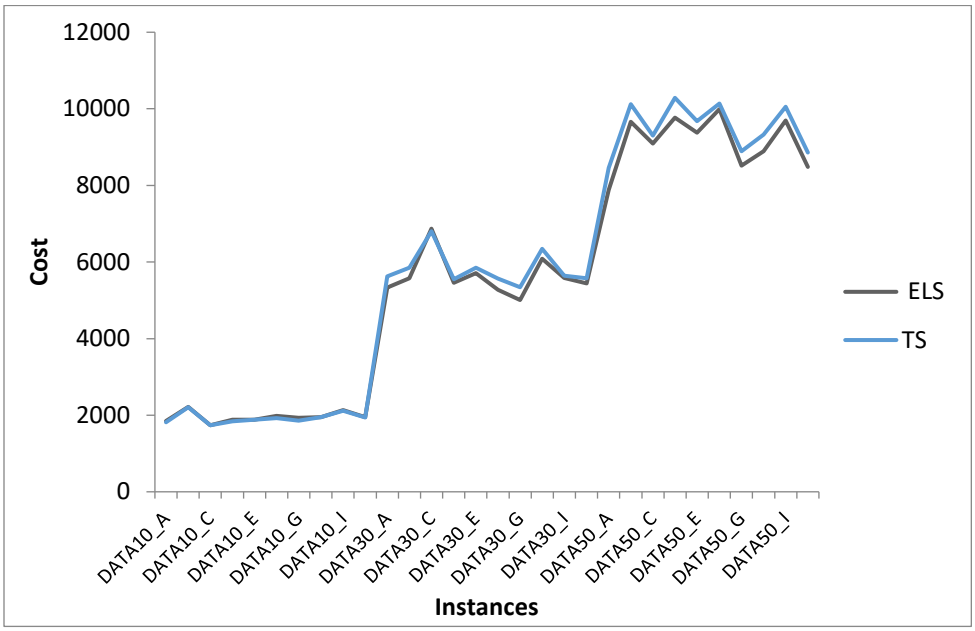

Figure 4: Cost comparison between ELS and TS

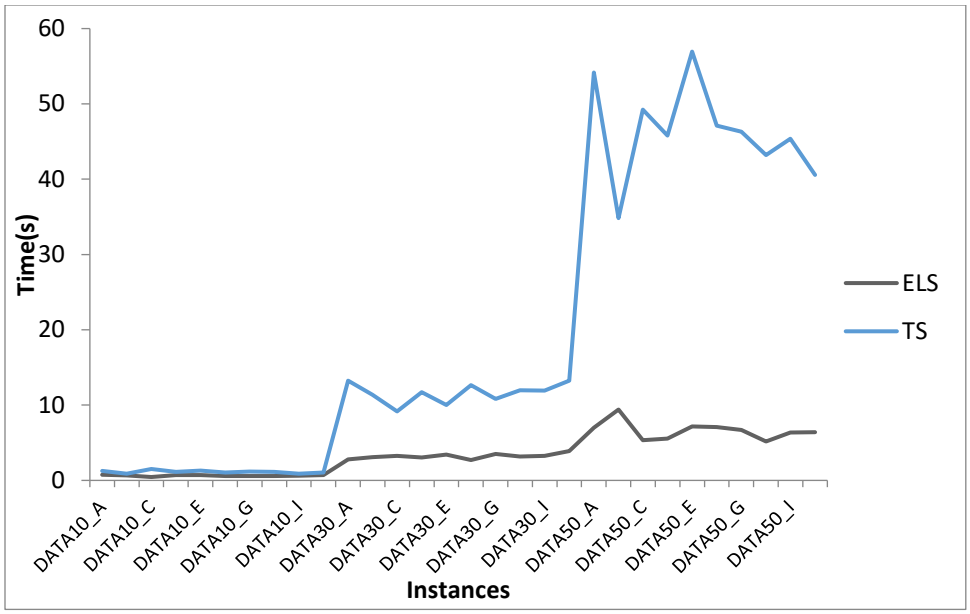

Figure 5. Computational time comparison between ELS and TS

From Figure 4 and 5, it is clear that ELS outperforms TS both on the quality of solutions and the time needed to find them. The performance of ELS is mainly related to its diversification procedure since both Metaheuristics are based on the same local search moves and strategy.

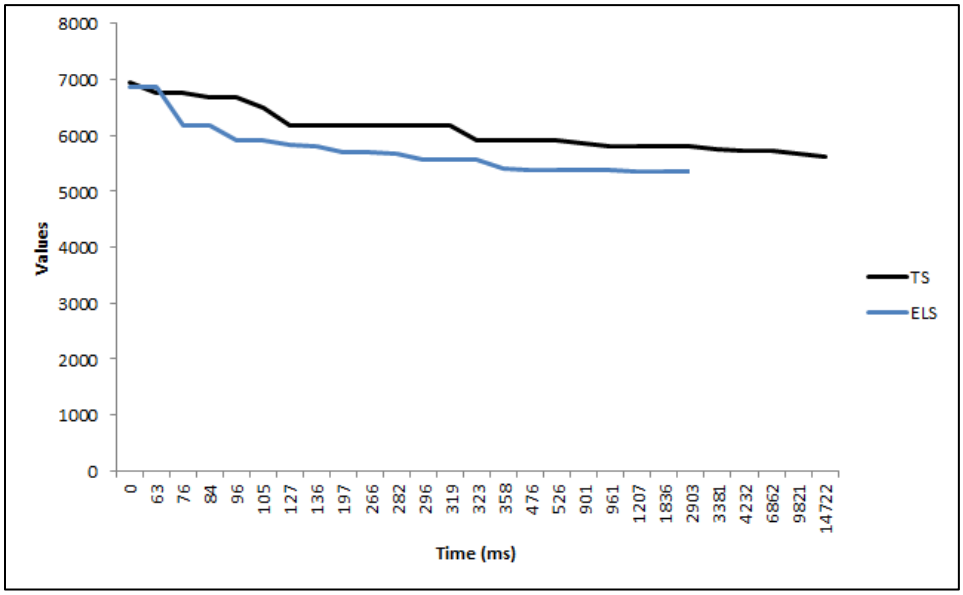

Figure 6: Convergence of ELS and TS on instance DATA30_A.

Figure 6 illustrates the convergence of ELS (upper curve) and TS (lower curve) on instance DATA30_A. ELS converges faster than TS, and shows strong improvements. This confirms the conclusions deduced from the two previous figures.

Analyzing overall solutions for the VRPCD, several conclusions can be drawn:

The results show that the metaheuristics perform much better than the BIH. They are able to improve strongly its results for all instances, especially when the size of problems becomes more important (over 30 customers), the GAP exceeds 35% within reasonable computational time especially for the ILS and ELS.
For all instances ELS converges faster and very good solutions were obtained, TS needs a very important CPU to find a good solution.

- Branch and cut algorithm

For some small instances, the VRPCD is solved to optimality, for the remaining ones the optimality was not achieved in a reasonable time (a time limit of 2 hours was used as a stopping criterion) . Table 3 shows the solutions obtained, the first column is the file name, the second is the size of the problem, and results are given in the last column. We note that Cplex was not able to find a solution for large problems because of the large number of variables and constrains.

Table 3.Results of the branch and cut algorithm

FILE	CLIENT	Branch & Cut	
		Cost	Time
DATA10_A	20	1832,69	7200
DATA10_B	20	2204,29	7200
DATA10_C	20	1737,64	7200
DATA10_D	20	1878,91	7200
DATA10_E	20	1880,27	3346,32
DATA10_F	20	2006,62	7200
DATA10_G	20	2021,68	7200
DATA10_H	20	1901,34	7200
DATA10_I	20	2209,71	7200
DATA10_J	20	1949,74	7200

6. Conclusion

Although cross-docking has been widely practiced within both manufacturing and retailers and brings benefits to companies, there are very few studies on the integration of vehicle routing problems and cross-docking. In this study, we have considered the VRPCD in which goods from suppliers and customers must be consolidated at a cross-dock before being dispatched to the customers. The problem was solved by means of a heuristic and three metaheuristics: A best insertion heuristic, a tabu search, an iterative local search and an evolutionary local search. Very good results are obtained. All metaheuristics found optimal solutions for small instances and each one of them has been successful to greatly improve the result of the constructive heuristic. Nevertheless, the ELS have been the best method among the three proposed; the ELS had the strength of finding good solutions with an excellent computational time. This proves the effectiveness of this algorithm for this type of combinatorial optimization problems. In future works, additional constraints can be considered, such as direct connections (suppliers to clients without going through the cross dock) when it is gainful. A lower bound will be also developed to better evaluate the performances of the proposed methods.

© IEOM Society International

References

Bartholdi, J. J., & Gue, K. R. The best shape for a crossdock. Transportation Science, 38:235–244. 2004

F.A Santos,G.R Mateus, and A.S Da Cunha.The pickup and delivery problem with cross docking. Comput Oper Res 40(4):1085-1093. 2013

Gumus, M.,&Bookbinder, J. H.Cross-docking and its implications in location distribution system. Journal of Business Logistics, 25, 199–229. 2004

H.R. Lourenço, O. Martin, and T. Stützle.Iterated local search. In F. Glover and G. Kochenberger, editors, Handbook of metaheuristics, pages 321–353. Kluwer. 2002

Larioui .S, Reghioui.M, Elfallahi, A, Elkadiri, K. A memetic algorithm for the vehicle routing problem with cross docking International Journal of Supply and Operations Management; 2.3: 833-855. 2015

Larioui.S, Reghioui. M. Resolution of the vehicle routing problem with cross-docking Proceedings of the International Conference on Industrial Engineering and Operations Management Rabat, Morocco, April 11-13, 2017

Lee, Y.H., Jung, J.W., Lee, K.M., 2006. Vehicle routing scheduling for cross-docking in the supply chain, Computers & Industrial Engineering, 51:247-256.

Lourenço, H. R., Martin, O. C., & Stutzle, T. (2010). Iterated local search: Framework and applications. In Handbook of metaheuristics. In F. Glover, G. Kochenberger, & F. Hillier (Eds.). International series in operations research and management science (Vol. 57, pp. 363–398). Springer.

Maknoon,Y,Laporte.G(2017). Vehicle routing with cross-dock selection. Computers & Operations Research, 7: 254-266

M.Wen, J.Larsen, J.Clausen, J.F Cordeau, and G Laporte, 2007,Vehicle routing with Cross-Docking, Journal of Operational Research Society , 60:1708–1718.

S. Wolf and P. Merz (2007).Evolutionary local search for the super-peer selection problem andthe p-hub median problem. In T. Bartz-Beielstein et al., editor, Hybrid Metaheuristics, number 4771 in Lecture Notes in Computer Science, pages 1–15, Berlin,. Springer.

Solomon, M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper.Res.35:254–265.

Stützle, T. (2006).Iterated local search for the quadratic assignment problem. European Journal of Operational Research, 174(3):1519–1539.

Tarantilis.C,D,Stavropoulou.F,Panagiostic P.Repoussis,(2012).Atabu search algorithm for consistent vehicle routing problem,Expert systems with applications 39;4233-4239

Vansteenwegen, P., Souffriau, W., VandenBerghe, G., & Van Oudheusden, D. (2009).Iterated local search for the team orienteering problem with time windows.Computers and Operations Research, 36

© IEOM Society International

Biographies:

Sanae LARIOUI was born in tangier,Morocco, on August 20, 1988, she received the Engineer Degree in logistics and transportation from the National school of applied sciences, Morocco, 2012. She graduated from from the Faculty of Sciences, Abdelmalek Essaadi University in Tetouan/ Morocco in logistics and optimization (Doctoral Thesis, 2019). Her research interests include optimization of industrial and logistics systems. She is professor in industrial engineering department and mumber of LSCD Research Laboratory (Laboratory system control and decision)

Mohamed REGHIOUI was born in tangier,Morocco,on 1982 .He received the engineer Degree from the polytechnic school of the University of Tours.he graduated from the University of Technology of Troyes(Doctoral theisis,2008). Currently, he is Professor at University Abdelmalek Essaadi. His research interests include optimization of industrial and logistics systems

Abdeĺlah El fallahi currently works at the department of logistics and applied mathematics, Abdelmalek Essaâdi University. Abdeĺlah does research in Computer Graphics, Artificial Neural Network and Algorithms. Their current project is large scale optimization problems and data mining

Rim MARAH, Professor in Computer Science Department, in ENSIT engineering school. Coordinator of LSCD Research Laboratory. My domains of interest are Smart Grid, Verification, Formal verification, Formal Methods and Smart Homes. My disciplines are Computer communications, Algorithms and Distributed Computing.

© IEOM Society International