Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Hybrid Heuristic Approaches For a Multi-Production Forward/Reverse Logistics System

Liu Changshi, Lai Mingyong

Economics and trade college, Hunan University, P.R.China, 410079
(Tel: 86-731-8684625; e-mail: liulongstone964@ 126.com)

Abstract: We consider a multi-production two-stage forward/reverse logistics system design problem where a fixed number of capacitated distribution/reclaiming centers are to be located with respect to capacitated suppliers and retail locations while minimizing the total costs, and take the random of demand/reclaiming into consideration. We also provide hybrid heuristic procedures for the solution of the problem, and develop transshipment heuristic to improve the duration of the hybrid approaches. Finally we present extensive computational results that show the high performance and effectiveness of the solution approaches.

1. INTRODUCTION

In general, a production forward/reverse logistics system design involves the determination of the best configuration, regarding location and size of the plants and distribution/reclaiming centers, their technology content, and product offerings and transportation decisions. In this paper, we consider a multi-production two-stage forward/reverse logistics system design problem where a fixed number of capacitated distribution /reclaiming centers are to be located with respect to capacitated suppliers and customers while minimizing the total costs. DCs can be seen as breakbulk and consolidation locations where the consolidated product shipments from plants are sorted and combined for shipment to retail locations to satisfy the demand at these sites.

Under these assumptions, we are interested in locating P DCs/RCs so that the total fixed location costs at the second level and the transportation costs at the stages is minimized without violating the capacity restrictions at the plant and DC levels. Thus, the problem considered in this paper involves strategic DC/RC location decisions and, mostly tactical, retailer-to-DC and DC-to-plant assignment decisions. Considerable attention has been devoted to these models for the location of plants and DCs. Several comprehensive reviews have been published in the last years (Geoffrion,et al.,1995;Erenguc,et al., 1997).

This paper makes several contributions to the existing literature. First, we consider a two-stage multi-production forward/reverse logistics system design problem, and handle the random of demand/reclaiming. Second, we develop customized meta-heuristic approaches with memory components. Third, the mathematical model reflects the potential cost savings that are due to economies of scale. The heuristic solution approaches can easily accommodate such changes in models and are equally applicable under these operational characteristics.

2.RELATED LITERATURE

Solution approaches related to our problem are optimization algorithms within the framework of Benders' decomposition (Geoffrion, 1974), heuristics based on branch-and-bound (Kaufman,et al., 1977; H. Ro,et al., 1984), and Lagrangian relaxation (Pirkul and Jayaraman, 1996, 1998).

However, these techniques consume extensive amounts of time and effort in finding optimal solutions for realistically sized problems. On the other hand, in (Syarif,et al., 2002; Keskin,et al., 2007), the genetic algorithms are shown to be effective in solving relatively large size problems, respectively.

How ever, the common ground of above approaches is that they do not take reverse logistics into consideration, not to mention handling the random of damand/reclaiming. Our work integrates some assumptions of (Geoffrion,et al.,1974; Pirkul,et al., 1998) with (Keskin,et al., 2007). Geoffrion and Graves develop and implement a solution technique. There are several distinctions between the assumptions in (Geoffrion,et al., 1974) and our study. The first distinction is that Geoffrion and Graves include the restriction that customers cannot be served by more than one DC (single-sourcing). Second, they employ four-index variables to represent the flow of products from plants to retailers through DCs. In our study, we employ seven-index variables for both stages that are connected via flow conservation at the DCs. Furthermore, in (Geoffrion,et al., 1974), each plant has a specific capacity dedicated to each product. In our work, we assume that each plant has an overall flexible capacity that can be utilized for different products via switch-overs.

H.Pirkul and Jayaraman (1998) consider locating both capacitated plants and DCs by heuristic based on Lagrangian relaxation. However, some of their model assumptions are similar to ours. First, the capacity restrictions at the plants are not dedicated for each product. Second, they consider multi-sourcing on the transportation links (plants to DCs and DCs to customers) as in our model. Third, they restrict, with an upper bound, the number of facilities and plants to be located.

Burcu B. Keskin and Halit Uster consider a two-stage production/distribution system design problem. However they only take the forward logistics into consideration.

3.THE MODEL

We consider a multi-production two-stage forward/ reverse logistics system, in the first stage, the products are transported from plants to capacitated distribution centers (DCs); and, in the second stage, they are transported from DCs to retailer or customer. At the same time, some used products or returned products are transported from retailer or customer to reclaiming centers (RCs). For simplicity, the DCs are the RCs. We assume that product mixes and capacities at established plant locations as well as the demand for different products at retail locations are known, and there exists a set of DCs/RCs to be located with capacity limitations. There are multi-commodities flowing, the demand at the retailers is constant, it is specific for each retailer and each product, the demand quantity and the reclaiming quantity are independent variables.

In this setting we determine how much of each product to transport from plants to DCs and from DCs to retailers, and how much reclaiming to transport from customers to RCs in such a way that the total transportation costs are minimized.

We use the following notation in a mixed integer programming formulation of our problem.

I set of customers, $i = 1,\dots, m$

J set of potential DCs, $j = 1,\dots, n$

K set of plants, $k = 1,\dots, f$

L set of products, $i = 1,\dots, g$

P number of DCs to be located

D_{il} demand at customer $i \in I$ for product $l \in L$

D_{il}^{r} reclaiming at customer $i \in I$ for product $l \in L$

W_j capacity limit at DC $j \in J$

W_j^{r} capacity limit at reclaiming center in DC $j \in J$

B_k capacity limit at plant $k \in K$

c_{ijl} unit transportation cost of product $l \in L$ from DC $j \in J$ to customer $i \in I$

t_{jkl} unit transportation cost of product $l \in L$ from plant $k \in K$ to DC $j \in J$

r_{ijl} unit transportation cost of reclaiming product $l \in L$ from customer $i \in I$ to DC $j \in J$

u_{lj}^{r} unit disposal cost of product $l \in L$ in DC $j \in J$ while reclaiming quantity exceeds the capacity of reclaiming center

u_{lj}^{s} unit opportunity cost of product $l \in L$ in DC $j \in J$ while demand be satisfied partially

u_{lj}^{c} unit storage cost of product $l \in L$ in DC $j \in J$ while distribution quantity exceeds the demand

We also define three sets of decision variables:

z_j 1 if DC and reclaiming center at location j is used, 0 o.w., $j \in J$

x_{ijl} amount of product $l \in L$ transported from DC $j \in J$ to customer $i \in I$

y_{jkl} amount of product $l \in L$ transported from plant $k \in K$ to DC $j \in J$

x_{ijl}^{r} amount of product $l \in L$ transported from customer $i \in I$ to DC $j \in J$

m_{ijl}^{d} amount of disposal product $l \in L$ while reclaiming quantity exceeds the capacity of reclaiming center

m^{s}_{ijl} amount of the lack of product $l \in L$ while demand be satisfied partially

m^{c}_{ijl} amount of the surplus product $l \in L$ distribution quantity exceeds the demand

Then, the problem can be formulated as follows:

$$\sum_{i\in I}\sum_{j\in J}\sum_{l\in L} c_{ijl}x_{ijl}z_j + \sum_{j\in J}\sum_{k\in K}\sum_{l\in L} t_{jkl}y_{jkl}z_j + \sum_{i\in I}\sum_{j\in J}\sum_{l\in L} r_{ijl}m_{ijl}z_j + \sum_{j\in J}\sum_{l\in L} u^{c}_{lj}m^{c}_{ijl} + \sum_{j\in J}\sum_{l\in L} u^{r}_{lj}m^{d}_{ijl} + \sum_{j\in J}\sum_{l\in L} u^{s}_{lj}m^{s}_{ijl}$$

S.t.

$$\sum_{j\in J} x_{ijl} = D_{il}, \sum_{j\in J} m_{ijl} \le D^{r}_{il}, \forall i \in I, \forall l \in L \quad (1)$$

$$\sum_{i\in I}\sum_{l\in L} x_{ijl} \le w_j z_j, \sum_{i\in I}\sum_{l\in L} m_{ijl} \le w^{r}_{j} z_j, \forall j \in J \quad (2)$$

$$\sum_{i\in I}\sum_{l\in L} x_{ijl} = \sum_{k\in K}\sum_{l\in L} y_{jkl}, \forall j \in J \quad (3)$$

$$\sum_{j\in J}\sum_{l\in L} y_{jkl} \le B_k, \forall k \in K \quad (4)$$

$$\sum_{j\in J} z_j = \mathrm{P} \quad (5)$$

$$x_{ijl} \le s_{ijl}z_j, \forall i \in I, \forall j \in J, \forall l \in L \quad (6)$$

$$\sum_{l\in L} y_{jkl} \le R_{jk}z_j, \forall j \in J, \forall k \in K \quad (7)$$

$$x_{ijl} \ge 0, y_{jkl} \ge 0, m_{jkl} \ge 0, \forall i \in I, \forall j \in J, \forall k \in K, \forall l \in L \quad (8)$$

$$z_j \in \{0,1\}, \forall j \in J. \quad (9)$$

In the objective function, the first term represents the costs from the DCs to the customers, and the second term is the costs from the plants to the DCs, the third term is the reverse costs from customers to the RCs, the fourth term represents the storage cost of superfluous product in DCs while distribution quantity exceeds the demand quantity, the fifth term represents the disposal cost of superfluous product in RCs while reclaiming quantity exceeds the capacity of reclaiming center, and the sixth term is the opportunity cost of the lack of demand in DCs.

Constraints (1) ensure that demand for each product l at each customer i is satisfied, and the reclaiming for each product l is less than the capacity limitation of RC j. Constraints (2) and (4) ensure the capacity restrictions at the DCs, RCs and the plants, respectively. There is no specific capacity requirement for each product at the plants and DCs. Constraints (3) are for flow conservation at each DC. Constraint (5) specifies the number of DCs to be located. In constraint set (6), s_{ijl} is given by min{ D_{il}, W_j } for $i \in I, j \in J$ and $l \in L$. These constraints state that a customer can only be assigned to an open DC. In constraint set (7), R_{jk} is given by min{ W_j, B_k } for $j \in J$ and $k \in K$. They are added to obtain a stronger formulation and are implied by constraints (2), (4) and (5). Finally, constraints (8) and (9) are non-negativity and integrality constraints.

4. HYBRID HEURISTIC SEARCH APPROACHES

In order to determine the set of open DCs/RCs, we use scatter search and tabu search meta-heuristics in our solution procedures. Scatter search has been implemented in various optimization problems (M. Laguna et al., 2003).

Our hybrid search heuristic operates on a binary vector z of size n, which represents the locations of the DCs/RCs. Given the locations of the DCs/RCs, i.e., an instance of z, we solve the corresponding capacitated transshipment problem to optimality. For a given solution vector z, we find its objective function value Z by adding the optimal objective value of the capacitated transshipment problem. The parameters used in the scatter search are the population size $2*h_{\max}$ where $h_{\max}$ is the diversification parameter, and the number of quality and diverse solutions in the reference set, which are denoted by b_1 and b_2, respectively. The selection of the parameters has an important effect on the performance of the scatter search. We randomly select our initial seed solution, a binary vector z, as an input to the diversification generation method where we generate a population of diverse solutions.

Step 1: Diversification. This step aims to produce trial solutions that differ from each other as significantly as possible. We summarize the ideas behind the procedure that we select in our implementation (Glover, 1997). Let z_i be the elements of a binary solution vector z. Given z as the seed solution, the generator creates $2*h_{\max}$ new solutions. According to Burcu B. Keskin and Halit Uster, the first $h_{\max}$ solutions $z_1,\ldots,z_{h_{\max}}$ are generated using an incremental parameter k and the following expressions: $z^{1+hk}_{h} = 1 - z^{1+hk}$, $k = 0,1,\ldots,\lfloor n/h \rfloor$. The maximum value of the incremental variable k changes for each value of h. The second $h_{\max}$ solutions are obtained using the relation $1 - Z_h$ where h = 1,…,hmax.

Step 2: Feasibility. The trial solutions are generally infeasible due to violations of constraints (2) and constraint (5). First, it is likely some of these initial solutions will not satisfy constraint (5) since the number of open facilities can be lower or higher than the prespecified P-value. The simple rule to correct this infeasibility is to open facilities randomly among the unused ones if $p > \sum_j Z_j$ and, otherwise, to close facilities randomly among the open ones until the total number of open facilities is equal to P . Second, we need to consider the feasibility of a trial solution in terms of the total capacity it provides. If the total capacity provided is less than the total demand, we open a facility chosen randomly among the unused ones and close a randomly chosen open facility. We swap only a pair of facilities between the open and unused facility groups until the total capacity feasibility is obtained. Therefore, constraint (5) is not violated. We continue to correct infeasibilities due to constraints (2) and (5) until all of the trial solutions in the population are feasible.

Step 3: Reference set construction. The reference set contains a set of good quality and most diverse solutions. The b_1 quality solutions are selected based on the objective function values, i.e., the solutions with the lower total cost values are selected as the quality solutions. In order to find b_2 diverse solutions, we use the distance between two solutions as a diversity measure. We first include the quality solutions in the reference set. Then, in the population, we look for a solution that is not currently in the reference set and that maximizes the minimum distance to all solutions currently in the reference set. We include that solution in the reference set as the most diverse solution. We continue in this manner until we have chosen a total of b_2 diverse solutions.

Step 4: Subset generator. We generate subsets of the reference set solutions that will later be combined into new solutions. We use three subset generators to obtain subsets of size 2, 3 and 4, known as Type II, Type III and Type IV, respectively. Subsets of Type II correspond to the pairs of solutions taken from the reference set. A Type III subset is formed by including the best quality solution and a pair of other solutions from the reference set. The third subset generator, Type IV, selects the two highest quality solutions from the reference set and joins them with pairs of the remaining solutions.

Step 5: Solution combination. The trial solutions in each subset that are generated in the previous steps are combined into a new solution by taking a linear combination of the solutions it includes. We use the reciprocal of the objective values of the solutions as coefficients of the trial solutions. Therefore, a solution with a lower objective value receives a higher weight. Clearly, an element of a combined solution can be a fractional value. In order to obtain a legitimate solution, we round each element of the final solution to its closest binary value.

Step 6: Reference set update. After obtaining a new solution, we determine whether this solution should be included in the reference set. First, if the objective value of the new solution is better (i.e., lower objective value) than any of the quality solutions in the reference set, we drop the worst solution among the quality solutions and add the new solution to the reference set. If this is not the case, we check the diversity of the new solution. We calculate the distance between the new solution and each solution in the reference set. If it is more diverse (the most distant) than any of the diverse solutions in the reference set, we drop the worst solution (with respect to the distance criteria) among the diverse solutions and add the new solution to the reference set. Finally, if the new solution does not satisfy these two conditions, then it is not eligible to be a member of the reference set. In which case, we move to the next subset to generate another new solution. If the reference set is updated, the algorithm goes back to Step 4 to generate new subsets. When the reference set cannot be updated further, the scatter search procedure is terminated. We implement a tabu search heuristics to improve the result.

Step 7: Improvement with tabu search meta-heuristics. Tabu search provides an opportunity to escape local optima and explore a larger subset of the solution space(Sun, 1998; Gendreau,at el,. 1994).

As before, we represent a solution to our problem with a binary vector $Z = (Z1,..., Zn)$ where $Zj \in \{0,1\}, j \in J$. Since tabu search uses a pair-exchange neighborhood over the sets of opened and unopened DC given by β and ϕ , respectively. A move in this neighborhood corresponds to simultaneously changing the z_j value for a $j \in \beta$ to zero and the z_j value for a $j \in \phi$ to one.

Tabu move restrictions are employed to prevent cycling and re-visiting previously visited solutions. We classify a solution obtained by a pair-exchange as a tabu if it corresponds to closing a DC which was opened in a recent accepted solution in the course of the procedure. The recency refers to the fact that tabu status is not permanent for an open DC. We employ a tabu tenure that is equivalent to the number of iterations an open DC remains a tabu. In this study, we choose the fixed tabu tenure approach.

The tabu search algorithm uses a tabu list T where $T = (T_1,..., T_n)$ with each T_j representing the tabu tenure of DC $j, j \in J$. If $T_j > 0$ for some $j \in J$, then the DC j is tabu. Any DC with a corresponding entry that is equal to zero in the tabu list is non-tabu. At each iteration, when a candidate solution results in opening a new DC j , T_j is assigned the tabu tenure and all other positive entries in the tabu list are decreased by one. This is a type of recency-based, or short-term memory, since the tabu list shows how recently the solutions were visited.

An aspiration criterion is used to overrule the tabu restrictions so that we can avoid escapes from attractive unvisited solutions. The aspiration criterion states that if a solution involving a tabu move has a better objective value than the best known solution, then the tabu status is disregarded. Otherwise, if the aspiration criterion is not satisfied, of course we continue to the next iteration with the best non-tabu solution.

4.1 Solving problem with two-stage transshipment heuristic

To improve the duration of the overall heuristic, we develop a two-stage transshipment heuristic for this problem after DCs being located by above heuristics. The heuristics help to reduce the total duration of the algorithm at least by half as reported in Section 5. Let $\hat{j} \in J$ be the set of open DCs so that $\left|\hat{j}\right| = P$. In this heuristic, we first solve the transportation problem between the DCs and the customers. Then, using the implied shipment requirements at the DCs as the demands for the DCs, we solve the transportation problem between the plants and the DCs. The algorithm of the two-stage transhipment heuristic is given in Display 1.

Display 1. Pseudocode of the transshipment heuristic

Input: $\hat{j}, c_{ijl}, t_{jkl}, i \in I, j \in \hat{j}, k \in K, l \notin L$

Output: $x_{ijl}, y_{jkl}, m_{ijl}, m^d_{ijl}, m^s_{ijl}\ m^c_{ijl}$

Sort c_{ijl} in an increasing order.
  for each c_{ijl} do
    if $D_{il} > 0$ then
      if $W_j > D_{il}$ then
        $x_{ijl} \leftarrow D_{il}, W_j = W_j - D_{il}, D_{il} \leftarrow 0$
      else
        $x_{ijl} \leftarrow W_j, D_{il} = D_{il} - W_j, W_j \leftarrow 0$
      end if
    end if
    if $D^r_{il} > 0$ then
      if $W^r_j > D^r_{il}$ then
        $m_{ijl} \leftarrow D^r_j, W^r_j = W^r_j - D^r_{il}, D^r_{il} \leftarrow 0$
      else
        $m_{ijl} \leftarrow W^r_j, m^d_{il} = D^r_{il} - W^r_j, D^r_{il} \leftarrow 0$
      end if
    end if
  end for
  Calculate $F_{jl} = \sum_{i \in I} x_{ijl}, \forall j, \forall l.$
  Sort t_{jkl} in an increasing order.
  for each t_{jkl} do
    if $F_{jl} > 0$ then
      if $P_k > F_{jl}$ then
        $y_{jkl} \leftarrow F_{jl}, P_k = P_k - F_{jl},$
        $m^c_{il} = P_k - F_{jl}, F_{jl} \leftarrow 0$
      else
        $y_{jkl} \leftarrow P_k, m^s_{il} = P_k - F_{jl}, P_k \leftarrow 0$
    end if
  end if
end for

5. COMPUTATIONAL RESULTS

We employ C++ and traditional bit string code for solving the proposed problem. In order to test the performance and effectiveness of the heuristic approaches, we conduct a series of numerical studies as shown in Table 1 (Appendix A) on randomly generated problems ranging from small to moderately large sized ones according to schedule described above. We consider five sets of problems with 10 instances in each; thus we solve a total of 50 instances. We run the C++ algorithms on a Pentium IV 3.2 GHz machine with 1 GB memory.

We summarize of the results for the hybrid heuristic approaches in Table 2 (Appendix A). i.e., the scatter search heuristic with, and without, the tabu heuristic improvement. In this experiment, the scatter search parameters are $h_{\max} = 10, b_1 = 1, b_2 = 2$, and the number of restarts is 6, i.e., each instance is solved five times, each starting with a different seed solution. We pick the best of the six runs for each instance as the heuristic solution, and the total duration of the six runs is considered as the overall runtime of the heuristic. To show the effectiveness of the heuristic, we report the average optimality gap (%), maximum optimality gap (%), and the number of optimum solutions. To compare the speed of the hybrid heuristic approaches we provide, in each set, the average and maximum overall runtimes for the 10 problems. The percentage optimality gaps are given by [100 * (Heuristic Obj. Value - Optimal Obj. Value)/Optimal Obj. Value], and the solution times are reported in seconds.

For small and medium problem instances, data sets 1, 2 and 5, the average gap between the hybrid heuristic approaches with and without improvement is less than 1%. For large instances, data set 3 and 4, the average gap can be as high as 4.5% and the maximum gap as high as 6.5%. Moreover, in 50 of the 90 cases, we find the optimum result.

[insert table 1]

[insert table 2]

6. CONCLUSIONS

We investigate a multi-production two-stage forward/reverse logistics system design problem. Although there have been a number of studies on our problems, there have not been sufficient studies involving solutions taking the opportunity cost, disposal cost and storage cost into consideration to handle the random of demand/reclaiming . This paper originated in an attempt to design effective and efficient hybrid heuristics approaches for the proposed problem, including scatter search and tabu search. We present scatter search and tabu search algorithms that perform very well even on large size problems. The scatter search heuristic combined with the tabu heuristic results in better optimality gaps and a high number of optimal solutions, thus, proving to be an effective method for solving the proposed problem. In our computational analysis, we observe the scatter search solutions can be improved by tabu heuristic.

We also develop a transshipment heuristics to reduce the total duration of the algorithm, and it can be easily adapted by changing the parameter Pk to Pkl to solve the problem when the plants have product specific capacities.

REFERENCES

A.M. Geoffrion, R.F. Powers (1995). 20 years of strategic distribution-system design – an evolutionary perspective. *Interfaces*, 25, 105–127.

A.M.Geoffrion,G.W. Graves (1974). Multicommodity distribution system design by Benders decomposition. *Management Science*,20,822–844.

A. Syarif, Y. Yun and M. Gen (2002). Study on multi-stage logistic chain network: A spanning tree-based genetic algorithm approach. *Computers and Industrial Engineering*, 43, 299–314.

Burcu B. Keskin and Halit Uster (2007). Meta-heuristic approaches with memory and evolution for a multi-product production/distribution system design problem. *European Journal of Operational Research*, 182, 663–682.

F. Glover (1989). Tabu search Part I. *ORSA Journal of Computing*, 1, 190–206.

F. Glover (1990). Tabu search Part II. *ORSA Journal of Computing*, 2, 4–32.

F. Glover, M. Laguna (1997). Tabu Search, *Kluwer Academic Publishers*, Norwell, MA.

F. Glover (1997). A template for scatter search and path relinking, in: J.K. Hao, E. Lutton, E. Ronald,

M. Schoenauer, D. Snyers (Eds.). *Artificial Evolution, Lecture Notes in Computer Science*, 1363, 13–54.

H. Ro, D. Tcha (1984). A branch-and-bound algorithm for the two-level uncapacitated facility location problem. *European Journal Operational Research*, 18, 349–358.

H. Pirkul and V. Jayaraman (1996). Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Science*, 30(4), 291–302.

H. Pirkul and V. Jayaraman (1998). A multi commodity, multi-plant, capacitated facility location problem: Formulation and efficient heuristic solution. *Computers and Operations Research*, 25, 869–878.

L. Kaufman, M.V. Eede and P. Hansen (1977). A plant and warehouse location problem. *Operations Research Quarterly*, 28(3),547–554.

M. Laguna and R. Marti (Eds.) (2003). Scatter Search: Methodology and Implementations in C, *Kluwer Academic Publishers*, Boston, MA.

M. Gendreau, A. Hertz and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science*, 40 (10), 1276–1290.

M. Sun (1998). A tabu search heuristic procedure for solving the transportation problem with exclusionary side constraints. *Journal of Heuristics*, 3(4), 305–326.

S.S. Erenguc, N. Simpson and A.J. Vakharia (1999). Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 219–236.

Appendix A.

Table1. Data sets optimum solution durations for test problems

	K	J	Y	X	W	M	W_j^r	L	C	S	P
1	[3,6]	[10,20]	[40,50]	[38,45]	[40,45]	[5,10]	[5,10]	0	[2,17]	[0,5]	[1,5]
2	[3,6]	[10,20]	[50,60]	[55,65]	[60,65]	[5,10]	[5,10]	[5,15]	0	[0,5]	[2,6]
3	[3,6]	[30,50]	[70,90]	[70,85]	[75,85]	[8,12]	[5,10]	0	[0,15]	[3,7]	[3,9]
4	[3,6]	[30,40]	[50,60]	[50,65]	[70,80]	[8,12]	[5,10]	[0,15]	0	[3,7]	[3,9]
5	[3,6]	[15,25]	[40,45]	[45,50]	[40,48]	[1,6]	[5,10]	[5,10]	0	[3,7]	[0,1]

Table2. Results for hybrid heuristics approaches

Data set	Without tabu improvement					With tabu improvement				
	Ave.gap	Max.gap	Ave.time	Max.time	No.opt	Ave.gap	Max.gap	Ave.time	Max.time	No.opt
Scatter search										
1	0.00	0.00	305.2	402.8	10	0.00	0.00	401.3	602.5	10
2	0.26	.075	406.9	996.1	5	0.00	0.00	680.6	1153.2	8
3	4.51	6.53	2256.4	5536.3	1	0.46	0.85	3482.6	6573.1	2
4	1.59	3.02	2157.6	4963.7	6	.023	0.62	2833.4	5634.8	5
5	0.12	0.63	598.6	1025.8	7	0.00	0.00	765.2	1268.5	8
Two-stage transhipment heuristics										
1	0.04	0.25	60.1	116.3	8	0.00	0.00	69.5	146.8	10
2	0.86	2.64	186.4	301.2	2	0.18	0.64	234.6	634.7	4
3	1.43	3.28	387.3	826.4	1	0.59	1.44	528.3	968.6	1
4	0.96	1.43	1063.2	2013.5	0	0.75	1.12	1684.2	2563.1	0
5	0.00	0.00	98.6	265.3	10	0.00	0.00	171.5	368.4	10