GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES FOR A SINGLE PRODUCT NETWORK DESIGN MODEL

Mahdis Haghighi Gashti[1,]* & Hassan Javanshir[2]

[1]M.Sc, Dep. of Industrial Engineering, Islamic Azad University, South Tehran Branch, Iran
[2]Assistant Professor, Dep. of Industrial Engineering, Islamic Azad University, South Tehran Branch, Iran
***Email :** Mahdis_haghighi@Yahoo.com

ABSTRACT
We consider a two-stage supply chain which contains a production facility and distribution centers (DCs) that supply retailers demand. The objective is to locate distribution centers in the network such that the sum of fixed facility location, pipeline inventory, and safety stock costs is minimized. We use Greedy randomized adaptive search procedures (GRASP) to solve the model and compare their performance to that of a Lagrangian heuristic and genetic algorithms (GAs) developed in earlier work. The comparison represent that GRASP provides solutions which are similar quality to those from the Lagrangian heuristic. The results show that GRASP has significantly better solutions than GA with only binary vector and GA with only random keys.

Keywords: *Network design, Lead time, Safety stock, Greedy randomized adaptive search procedures (GRASP).*

1. INTRODUCTION

The distribution networks' design is one of the most important decision in supply chain (SC). This problem has been considered as a strategic decision that includes determining which distribution centers(DCs) must supply retailers demand [20]. Today competitive business environment increase and product life cycles being shorten so it makes the customer demands become more uncertain [1], [4]. The safety stock in cases which customer demands are stochastic, is influenced by lead time [5], [12], [21].

Sourirajan et al. [19] consider a facility location–allocation model that contains fixed facility location, the pipeline inventory and safety stock costs. The objective is to locate distribution centers at certain sites to serve groups of retailers. Each DC has limited capacity and must hold enough saftey stock to satisfy retailers demands. This problem is SPNDLS (a Single Product Network Design Model with Lead time and Safety Stock Considerations) which consist of lead time and safety stock. It is a non-linear integer programming formulation which recognized to be NP-hard was developed by Sourirajan et al. [19]. They obtain near-optimal solutions by using a Lagrangian heuristic. Sourirajan et al. [20] discuss the use of genetic algorithms (GA) for solving the SPNDLS problem of Sourirajan et al. [19].

In this paper, we discuss the use of Greedy randomized adaptive search procedures (GRASP) for solving the SPNDLS problem of Sourirajan et al. [19] and compare their performance to that of a Lagrangian heuristic and genetic algorithms(GAs). The advantage of GRASP is that the CPU times for large problems required by GRASP are much lower than that of the GAs. Furthermore the results show that GRASP has significantly better solutions than GA with only binary vector (GABV) and GA with only random keys (GARK).

We review literature of the problem in Section 2. In Section 3 we describe the SPNDLS problem and the formulation which is explore by Sourirajan et al. [19]. In Section 4 we explore GRASP and the algorithm to solve SPNDLS. In Section 5 we present the computational experiments and the comparison of GRASP and Lagrangian heuristic and genetic algorithms(GAs) developed in earlier work. Finally in Section 6 we explain the conclution and som future directions.

2. LITERATURE REVIEW

The objective of the Uncapacitated Facility Location Problem (UFLP) is to locate DCs among the candidate locations to serve the demand points while minimizing the sum of fixed location and transportation costs. The objective and constraints are linear but the problem is NP-hard [19]. The Capacitated Facility Location Problem (CFLP) has the same objective as the UFLP, with the addition of capacity constraints that limit the demand that can be served by each candidate location [20].

Some authors present models and solution procedures for the CFLP and its variations. However, these models ignore lead times and service levels. In some cases they consider the effects of safety stock but do not address lead time and vice versa. The detail of this literature presents by Sourirajan et al. [19].

Sourirajan et al. [19], Liu and Zhang [13] and Park et al. [16] consider lead times and safety stocks in network design. They study the effects of resource utilization on lead times and safety stock risk pooling benefits. In these

cases Lagrangian heuristics used to obtain near-optimal solutions with reasonable computational requirements. Sourirajan et al. [20] explored the use of GAs based on binary vector and random keys encodings for solving the SPNDLS and illustrated the advantages of GAs over the Lagrangian heuristic for such problems. In this paper we explain the use of Greedy randomized adaptive search procedures (GRASP) for solving the SPNDLS problem of Sourirajan et al. [19] and compare their performance to that of a Lagrangian heuristic and genetic algorithms(GA).
We present the SPNDLS problem description and formulation of Sourirajan et al. [19] in the next section.

3. PROBLEM DESCRIPTION AND FORMULATION

Sourirajan et al. [19] considere a distribution network design problem for a two-stage supply chain which contains a production facility and distribution centers (DCs) that supply retailers demand. The objective is to locate DCs to serve the retailers such that the sum of fixed location and inventory (pipeline and safety stock) costs is minimized. They assume that the retailers' demands are independent and follow a Poisson process [3], [18], [14], [15].
They assume that products are shipped from the production facility to a DC in full truckloads. For such a replenishment process, the replenishment lead time at a DC has three components:

1. Load make-up time – The time spent in the waiting area of the production facility before the products are sent to the DC.
2. Constant DC replenishment time (time/unit) – The replenishment lead time between the production facility and the DC due to the physical locations of facilities.
3. Congestion time – The time spent in the unloading zone. At high utilization of the resources at the unloading zone, shipments have to wait longer in the queue.

Sourirajan et al. [19] explain that due to the retailer' demand is stochastic and the lead time is invariable the safety stock at a DC given by, $(z_\alpha\sqrt{L\sum_{k\in K}\sigma_k^2})$, where $P(z \le z_\alpha) = \alpha$ which α is the service level that has to be achieved at the retailers. Therefore the retailers' demands follow Poisson process, the variance is equal to the mean. Consequently the amount of safety stock is $(z_\alpha\sqrt{L\sum_{k\in K}D_k})$.

3.1. SPNDLS formulation

In order to present the formulation of SPNDLS, Sourirajan et al. [19] define the following notation:

Inputs:

k	set of retailers ($k = 1,\dots,N$)
j	set of possible DC locations ($j = 1,\dots,N$) – same as the set of possible retailer locations
f_j	fixed cost of locating a DC at location j
C_j	capacity of the DC at location j
θ_j	unit cost of pipeline inventory for the DC at location j
H_j	unit cost of safety stock at a DC at location j
p_j	load make-up time parameter of lead time for a DC at location j
q_j	constant lead time component per unit for a DC at location j
r_j	congestion parameter of lead time for a DC at location j
D_k	mean demand at retailer k
β_j	adjusted holding cost per unit for a DC at location j
α	service level that has to be achieved at the retailers
z_α	inverse of the Standard Normal for a probability of α

Decision variables:

Y_j = 1 if a DC is built at location j, 0 otherwise
X_{jk} = 1 if retailer k assigned to DC at location j, 0 otherwise

The SPNDLS problem formulate by Sourirajan et al. [19] as follows. The details of the model can be found in the published paper.

$$\text{Min} \sum_j f_j Y_j + \sum_j \theta_j \left(p_j + q_j \sum_k D_k X_{jk} + \frac{r_j \sum_k D_k X_{jk}}{C_j - \sum_k D_k X_{jk}}\right) + \sum_j \beta_j \sqrt{\left(p_j + q_j \sum_k D_k X_{jk} + \frac{r_j \sum_k D_k X_{jk}}{C_j - \sum_k D_k X_{jk}}\right)} \quad (1)$$

$$s.t.\ X_{jk} \le Y_j \quad \forall j,k, \quad (2)$$

$$\sum_j X_{jk} = 1 \quad \forall k, \quad (3)$$

$$\sum_j D_k X_{jk} \le C_j \quad \forall j, \quad (4)$$

$$X_{jk} : \{0,1\} \quad \forall j,k, \quad (5)$$

$Y_j \; : \; \{0,1\} \qquad \forall j.$ (6)

4. GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES (GRASP) FOR THE SPNDLS

Hart and Shogan [11] proposed a multi-start approach based on greedy randomized constructions, but without local search which is called semi-greedy heuristic. GRASP (Greedy Randomized Adaptive Search Procedures) were introduced in 1989 by Feo and Bard [7], [8]. The GRASP metaheuristic is a multi-start or iterative process, where each iteration consists of two phases: construction and local search [9], [10]. The construction phase builds a feasible solution and local search phase investigate the neighborhoods to find local minimum. At each iteration of construction phase select candidate elements. The candidate set C is defined for each problem. The incremental costs, $c(e)$ for all $e \in C$ (the candidate elements) are evaluated (greedy evaluation function). Afterward the restricted candidate list (RCL) is built. For making RCL, let $c^{\min}$ and $c^{\max}$ be, respectively, the smallest and the largest incremental costs. The threshold value for elements in RCL is $(c^{\min} + \alpha(c^{\max} - c^{\min}))$ $\alpha \in [0,1]$. If the value of $c(e)$ is lower than or equal to the threshold value, the candidate element can be inserted to RCL. After making RCL an element from the RCL at random is selected. Finally the candidate set C is updated and the incremental costs $c(e)$ for all $e \in C$ are reevaluated. The local search replaces current solution with better solution which is in the neighborhood of current solution. For local search the neighborhood investigate in two ways: best-improving or first-improving strategy. In the best-improving strategy we consider all neighbors for finding best solution. In the case of first-improving strategy we consider the neighbors till find the first solution which the cost function value is smaller than the current solution [17]. In this paper we use the first-improving strategy duo to the computation times is smaller than the best-improving strategy.

4.1. Steps of GRASP for SPNDLS

Sourirajan et al. [19] developed the Lagrangian heuristic for finding near optimal solution. In this case they explain the steps for finding lower and upper bound of the problem.Afterward Sourirajan et al. [20] presented the genetic algorithm for the SPNDLS and it generate good result.

In this paper we present GRASP to find solution for SPNDLS. We use this metaheuristic due to its structure. The structure of GRASP for solving the problem in some steps, is similar to Lagrangian heuristic. It use both greedy heuristic and randomize algorithm in making construction phase. Then it search the nighbors to find first-improved solution. At each inetration the steps of the algorithm are below:

Construction phase:

Step 1: Sort the retailers in decending order of their mean demand and index the sorted list of retailers as $k = 1, \dots, N$. Set $k = 1$

Step 2: For kth retailers if the mean demand of retailer is lower than capacity of DC location j, then compute the greedy evaluation function $c(e)$ which is objective function of SPNDLS (1).

Step 3: Let $c^{\min}$ and $c^{\max}$ be, respectively, the smallest and the largest incremental costs. If the value of $c(e)$ is lower than or equal to $(c^{\min} + \alpha(c^{\max} - c^{\min}))$ then insert the DC location j to RCL

Step 4: Select a DC location j at random and set $X_{jk} = 1$ and $Y_j = 1$

Step 5: Set $k = k + 1$. If $k > N$ then STOP and compute objective function value f. Else go to step 1.

Local search phase

Step 6: While the first solution which objective function value f' is less f do the step 1 to step 5

5. COMPUTATIONAL EXPERIMENT

The problem data for GRASP is the same as that used for testing Lagrangian heuristic and GA algorithm [19], [20]. The data was derived using the 1990 census in Daskin (1995) [2]. The problem sizes are 15-node, 49-node, 88-node and 150-node which show the number of retailers. Each retailer locations are candidate for locating DCs. The capacity of DC is multiple of mean demand of retailer. We set the multiple equal to 3,4 and 5. The mean demand is equal to population for cities divided by 1000. In this case we use 97.75% service level and $z_\alpha = 2$. The fixed cost for locating DCs is derived in Daskin [2]. The constant lead time component per unit for a DC at location j, q_j is equal to 10. The load make-up time p_j and congestion parameter of lead time r_j derived from the results of Eskigun' work [6]. Let M denote the shipment size from plant to the DCs. We set M to the lowest mean demand among all retailers. Eskigun [6] set $p = (M - 1)/2$ and $r = C$ which C is the capacity of DCs. The problem data for unit cost of pipeline inventory θ_j and adjusted holding cost β_j were given in Ozsen [14]. The interation for GRASP in this experiment is set to 50 for 15-node and 49-node and 100 for 88-node and 150-node.

We use the following notations in our comparison

GABV: GA with only binary vector encoding (Sourirajan et al., 2009)

GARK: GA with only random keys encoding (Sourirajan et al., 2009)

GARKBV: GA with hybrid chromosome representation (Sourirajan et al., 2009)
Lag: Lagrangian heuristic (Sourirajan et al., 2007)
GRASP: Greedy randomized adaptive search procedures
Gap: estimate of the optimality gap for any problem instance given by $(\frac{(UB - LB)*100}{LB})$ where UB is the best solution found by the GRASP for that problem instance and LB is the final lower bound from Lag for that problem instance.
CPU: average CPU requirement per replication per instance (seconds)
In this paper we code the GRASP with Visual Basic sowftware and the result present in the following section.

5.1. Comparison of GRASP with Lagrangian heuristic and GA algorithm

In this section we compare GRASP with Lagrangian heuristic and GA algorithm for solving SPNDLS. We use the data and solution for Lagrangian heuristic in Sourirajan et al. [19] and GA algorithm in Sourirajan et al. [20].

Table 1. Average and worst-case gap for the GRASP

	Average gap					Worst-case gap				
	Lag	GRASP	GABV	GARKBV	GARK	Lag	GRASP	GABV	GARKBV	GARK
15-Node	0.34	0.36	0.48	0.33	1.34	2.15	2.14	2.67	2.07	5.89
49-Node	0.20	0.22	0.23	0.20	7.41	0.88	1.52	1.04	0.88	20.39
88-Node	0.82	0.85	0.77	0.79	45.05	2.47	5.76	2.31	2.31	130.06
150- Node	0.46	0.48	0.53	0.46	156.54	0.85	14.99	1.38	0.85	547.11
Overall (Avg/Max)	0.45	0.48	0.50	0.45	52.58	2.47	14.99	2.67	2.31	547.11

Table 1 shows GRASP is same quality as Lagrangian heuristic in average gap. This can explain by the structure of GRASP and the way that the Lagrangian heuristic makes lower bound and upper bound. In both algorithms we use greedy heuristic and improvement method. However in GRASP we use randomized algorithm. By comparing the three GA algorithms and GRASP, we conclude that GRASP has better solution than GABV and GARK in average gap. The table 1 shows that the GARKBV is better in average and worst case from others. GRASP in worst case gap has less quality from Lagrangian heuristic, GABV and GARKBV. The average gap for GRASP in this model for different size of problem is between 0.22%-85% and the worst case gap is between 1.52%-14.99%. Table 2 shows the CPU time for GRASP and other algorithms. It shows that in large size problem the GRASP has lower time than the others in both average and worst case. In small size problem the CPU time in GRASP is similar to GAs.
By running the problem with GRASP based on different multiple (3,4 and 5) it shows that the CPU times don't change due to this algorithm at first would check the capacity constraint and if it is satisfied, it makes the greedy evaluation function c(e). Consequently the capacity multiples do not affect the CPU time in GRASP. Sourirajan et al. [19] explained that in Lagrangian heuristic as available capacity increase the CPU time dose not change significantly. Sourirajan et al. [20] presented that in GA if available capacity increase then the CPU time increase.

Table 2. Average and worst-case CPU for the GRASP

	Average CPU					Worst-case CPU				
	Lag	GRASP	GABV	GARKBV	GARK	Lag	GRASP	GABV	GARKBV	GARK
15-Node	9	3	2	2	2	9	4	2	3	2
49-Node	65	30	30	32	16	80	30	43	52	16
88-Node	227	236	336	301	140	281	338	538	611	149
150- Node	674	574	1522	1527	486	831	580	2584	3194	531

6. CONCLUSION AND FUTURE DIRECTION

In this paper we explore the use of GRASP for SPNDLS. Sourirajan et al. [19] present the SPNDLS and use Lagrangian heuristic to find solution for this problem. Sourirajan et al. [20] explore the GA in SPNDLS. They prove that GA in small and medium sized problem has better quality than Lagrangian heuristic. However for large problem the CPU time of GA is higher than Lagrangian heuristic. In this paper we represent that GRASP has the same quality as Lagrangian heuristic but less quality than GARKBV. The GRASP has less average gap than GABV and GARK. The CPU times for different size of problem by using GRASP are lower than other algorithms.

In the future, we would like to consider the priority effect in GRASP. Sourirajan et al. [19] explain the priority as weight setting given to pipeline and safety stock. We also plan to develop the SPNDLS by considering stochastic lead time and shortage in the model.

7. REFERENCES

[1] Chen, M.S., Lin, C.T., Effects of centralization on expected costs in a multi-location newsboy problem. *Journal of Operations Research Society*. **40** (6), 597–602 (1989).

[2] Daskin, M.S., "Network and Discrete Location Models, Algorithms and Applications". Interscience Series in Discrete Mathematics and Optimization. Wiley, New York (1995).

[3] Daskin, M.S., Shen, Z.-J.M., Coullard, C., An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*. **110**, 83–106 (2002).

[4] Eppen, G., Effects of centralization on expected costs in a multi-location newsboy problem. *Management Science*. **25** (5), 498–501 (1979).

[5] Eppen, G.D., Martin, R.K., Determining safety stock in the presence of stochastic lead time and demand. *Management Science*. **34** (11), 1380–1390 (1988).

[6] Eskigun, E., Outbound supply chain network design for a large-scale automotive company. Unpublished Doctoral Dissertation, School of Industrial Engineering, Purdue University, West Lafayette, IN, USA (2002).

[7] Feo, T.A., Bard, J.F., Flight scheduling and maintenance base planning. *Management Science*. **35**,1415–1432 (1989).

[8] Feo, T.A., Bard, J.F., The cutting path and tool selection problem in computer-aided process planning. *Journal of Manufacturing Systems*. **8**,17–26 (1989).

[9] Feo, T.A., Resende, M.G.C., A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*. **8**, 67–71 (1989).

[10] Feo, T.A., Resende, M.G.C., Greedy randomized adaptive search procedures. *Journal of Global Optimization*. **6**, 109–133 (1995).

[11] Hart, J.P., Shogan, A.W., Semi-greedy heuristics: An empirical study. *Operations Research Letters*. **6**, 107–114 (1987).

[12] Karmarkar, U., "Manufacturing lead times, order release and capacity loading". In: Graves, S.C., Rinnooy Kan, A.H.G., Zipkin, P. (Eds.), Handbooks in Operations Research and Management Science, vol. 4. Elsevier, Amsterdam, pp. 288–291 (1993).

[13] Liu, X.F., Zhang, M., Cost Optimization Model of Distribution Systems in Supply Chain under Stochastic Demand. 2009 Chinese Control and Decision Conference (CCDC 2009).

[14] Ozsen, L., "Location-inventory planning models: Capacity issues and solution algorithms". Unpublished Doctoral Dissertation, Northwestern University, Evanston, IL, USA (2004).

[15] Ozsen, L., Coullard, C.R., Daskin, M.S., Capacitated warehouse location model with risk pooling. *Naval Research Logistics*. **55** (4), 295–312 (2008).

[16] Park, S., Lee, T.E., Sung, S.C., A three-level supply chain network design model with risk-pooling and lead times. *Transportation Research Part E*. **46**, 563–581 (2010).

[17] Resende, M.G.C., Ribeiro, C.C., Greedy randomized adaptive search procedures. AT&T Labs Research Technical Report TD-53RSJY, version 2 (2002).

[18] Shen, Z.-J.M., Coullard, C., Daskin, M.S., A joint location-inventory model. *Transportation Science*. **37**, 40–55 (2003).

[19] Sourirajan, K., Ozsen, L., Uzsoy, R., A single product network design model with lead time and safety stock considerations. *IIE Transactions*. **39** (5), 411–424 (2007).

[20] Sourirajan, K., Ozsen, L., Uzsoy, R., A genetic algorithm for a single product network design model with lead time and safety stock considerations. *European Journal of Operational Research*. **197**, 599–608 (2009).

[21] Yang, B., Geunes, J., Inventory and lead time planning with lead-time-sensitive demand. *IIE Transactions*. **39** (5), 439–452 (2007).