Available online at www.sciencedirect.com

SciVerse ScienceDirect

AASRI Procedia 1 (2012) 226 – 231

AASRI Procedia

www.elsevier.com/locate/procedia

2012 AASRI Conference on Computational Intelligence and Bioinformatics

Application of Shuffled Frog Leaping Algorithm to an Uncapacitated SLLS Problem

Zhao Liping[a], Wang Weiwei[b], Han Yi[b,]*, Xu Yefeng[b], Chen Yixian[b]

[a]*College of Information and Engineering，Jinhua Polytechnic, Jinhua 321017, China*

[b]*College of Economics and Management, Zhejiang University of Technology, Hangzhou 310023, China*

Abstract

Shuffled frog leaping algorithm (SFLA) is a population-based novel and effective meta-heuristics computing method, which received increasing focuses from academic and engineering optimization fields in recent years. Since SFLA is a combination of Memetic algorithm (MA) with strong local search (LS) ability and particle swarm optimization (PSO) with good global search (GS) capability, it is of strong optimum-searching power and easy to be implemented. This paper is devoted to investigate the ability of shuffled frog leaping algorithm (SFLA) to solve an uncapacitated single level lot-sizing (SLLS) problem. Through comparing with some heuristic algorithms, the result showed the feasibility and effectiveness of SFLA.

Keywords: Shuffled frog leaping algorithm; single level lot sizing problem; uncapacitated; heuristic

1. Introduction

The lot-sizing (LS) problem is the key production planning problem in materials requirements planning (MRP) systems. Its aim is to decide the optimal production lot size and the inventory volume to minimize the production cost, the inventory carrying cost, the back ordering cost etc. People used to adopt heuristics to solve MLLS problems (Silver et al., 1998). Recently, the applications of simulated annealing (SA) algorithm,

* Corresponding author. Tel.: +86-18957144810.
*E-mail address:*hanyi@zjut.edu.cn.

doi:10.1016/j.aasri.2012.06.035

genetic algorithm(GA) and PSO algorithm have been adopted to solve multilevel lot-sizing (MLLS) problems (Kennedy and Eberhart, 1995, Clerc, 2000, Dellaert and Jeunet, 2000, Han et al., 2009, Tang, 2004, Eberhart and Kennedy, 1995, Shi and Eberhart, 1998, Shi and Eberhart, 1998, Yelle, 1979, Dong et al., 2005). The single level lot-sizing (SLLS) problem is a special case of MLLS problem. Shuffled frog leaping algorithm (SFLA) is proposed by Eusuff and Lansey in 2003 (Eusuff and Lansey, 2003), which now has been the hotspot in intelligent optimization fields. Since SFLA is a combination of Memetic algorithm (MA) with strong local search (LS) ability and particle swarm optimization (PSO) with good global search (GS) capability, it is of strong optimum-searching power and easy to be implemented. So far, few literatures have introduced the application of SFLA to SLLS problem. Here we solve a single level lot-sizing problem with SFLA coded in C++ language in order to see whether or not SFLA is a feasible and effective tool for this kind of problem.

2. Mathematical Model of SLLS problem

We suppose that the planning span is limited to T, the demands d_t for each period in the span is pre-known, the production cost for single product will not change with the number of production volume, the lead time is 0, no backlogging is allowed, the setup cost for each lot is const and the inventory in the starting period and the last period is 0.

Nomenclature

S_t	the setup cost for the t^th^ period
Y_t	a binary variable (0-1), 1 representing one lot of products is processed in t^th^ period, otherwise 0
C_t	the production cost of one unit product
X_t	the amount of products produced in t^th^ period
h_t	the unit inventory carrying cost in t^th^ period
I_t	the inventory level at the end of t^th^ period
M_t	$\sum_{k=t}^{T} d_k$

$$Min.Z = \sum_{t=1}^{T}(S_tY_t + C_tX_t + h_tI_t) \quad (1)$$

$$S.t.X_t + I_{t-1} - I_t = d_t \,, t = 1,...,T \quad (2)$$

$$X_t \le M_tY_t \,, t = 1,...,T \quad (3)$$

$$Y_t \in \{0,1\}, t = 1,...,T \quad (4)$$

$$X_t, I_t \ge 0, t = 1,...,T \quad (5)$$

The objective function (1) represents the total cost is composed of production cost, setup cost and inventory carrying cost; equation (2) represents the balance equation of production inventory and demand; constraint (3) shows that the production amount in a period can not exceed the limit; constraint (4) is a binary variable; constraint (5) requires that the production amount and the inventory level is larger than or equal to 0.

3. SFLA

SFLA is a recently popular meta-heuristic based on the memetic evolution of a group of frogs when seeking for the location that has the maximum amount of available food. Proposed in 2003, SFLA is the combination of the merits of memetic algorithm (MA) and PSO. In SFLA, the population consists of a group of frogs (solutions) that is partitioned into subsets (memeplexes). Different memeplexes, each performs a local search, are considered as different cultures (memes) of frogs.

The SFLA is described as follows: assume that the initial population is formed by P randomly generated frogs. For L-dimensional problems (*L* variables), a frog i is represented as $X_i=(x_{i1}, x_{i2},., x_{iL})$. Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m memeplexes, each containing n frogs (i.e.$P = m \times n$). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the mth memeplex, and frog m+1 goes back to the first memeplex, etc.

Within each memeplex, the frogs with the best and the worst fitness are denoted by X_b and X_w. The frog with the global best fitness is identified as X_g. Here, a process similar to PSO is applied to improve X_w (not all frogs) in each small group. The adjusting measure is as follows:

$$S = rand() \times (X_b - X_w) \tag{6}$$

$$X_{new} = X_w + S, -S_{max} \le S \le S_{max} \tag{7}$$

Where *rand*() is a random number between 0 and 1. S_{max} is the maximum change in a frog's position. If a better solution X_{new} is produced, it replaces the worst frog X_w. Otherwise, the calculations in Eqs.(6) and Eqs.(7) are repeated by replacing X_b with X_g in Eqs.(6). If no improvement is achieved under this situation, then a new solution is randomly generated to replace X_w. Then the memeplex is reordered and renewed within local search times and thereafter all the memeplexes are shuffled together to exchange information and reallocated for next search process. The flowchart of SFLA can be given out in Fig 1 (Eusuff and Lansey, 2003).

3.1. Coding Method

According to the characteristics of SLLS problem, a solution is a one dimensional vector composed of a string of 0 or 1 with the length *T*, such as {1,0,0,1,0,1,1,0,0,0,1,0}. Since the demand for products in each period is known beforehand like {10,10,10,10,10,10,10,10,10,10,10,10}, the solution can be easily translated into a production decision {30,0,0,20,0,10,40,0,0,0,20,0}. Then the inventory level and inventory carrying cost can be decided accordingly.

3.2. Solution Adjustment Method

Since the basic SFLA is designed to solve the problems with continuous objective functions, the solution adjustment equations can not be directly applied to discrete SLLS problem. Then the way of adjusting a solution is revised into a crossover operator resembling the one adopted in genetic algorithm (GA). While producing a new frog (solution) in a memeplex, the best frog and the worst frog are crossed over to produce two new solutions. After comparison of the objective function value, the new solution with a better result is kept and the other one is discarded.

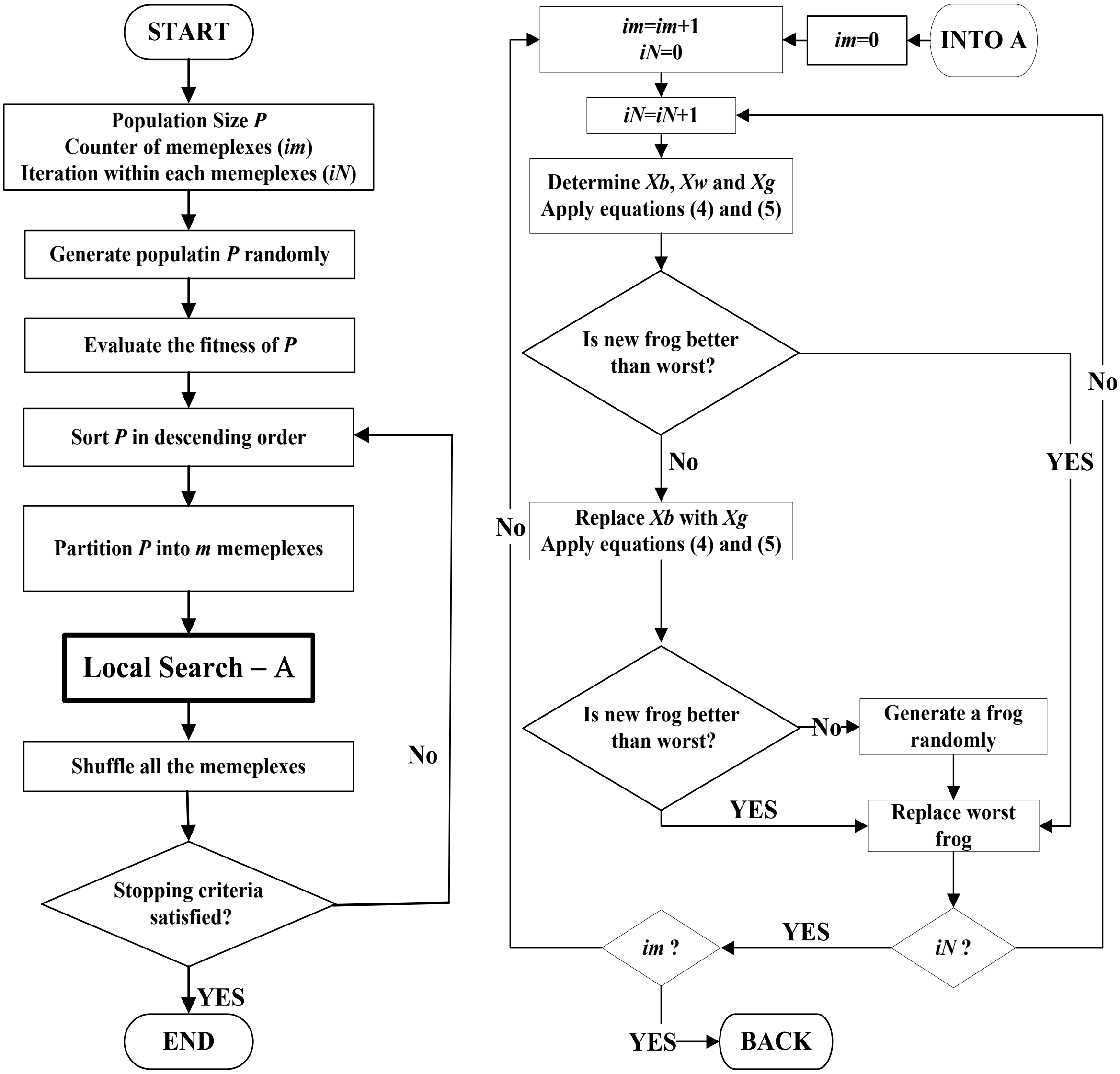
START
Population Size *P*
Counter of memeplexes (*im*)
Iteration within each memeplexes (*iN*)
Generate populatin *P* randomly
Evaluate the fitness of *P*
Sort *P* in descending order
Partition *P* into *m* memeplexes
Local Search – A
Shuffle all the memeplexes
Stopping criteria satisfied?
No
YES
END
INTO A
im=0
im=*im*+1
iN=0
iN=*iN*+1
Determine *Xb*, *Xw* and *Xg*
Apply equations (4) and (5)
Is new frog better than worst?
No
YES
No
Replace *Xb* with *Xg*
Apply equations (4) and (5)
No
Is new frog better than worst?
No
Generate a frog randomly
YES
Replace worst frog
im ?
YES
iN ?
YES
BACK

Fig. 1. Flow chart of SFLA

4. Experiment Result

Here, we tested SFLA with a single-level example with 1 item and 12 periods as shown in Table 1 and compared the result of SFLA with those of some heuristic algorithms, such as Wagner-Whitin(WW), Silver-

Meal(SM),EOQ, Lot-4-Lot and Least Unit Cost. External demand is randomly chosen with a mean value of 92. The max length of a velocity is set to be $0.02 \times M \times T$. Holding cost for inventory is set equal to $200 - 15i$ with i increases from 1 to 9. The set-up cost is set equal to $100i$ with i increases from 1 to 9. Some parameters in our algorithm are set to be constant over time. The crossover rate is 0.6 for PSO SFLA. After a pre-test, the population size is 100; the size of a memeplex is 10; the iteration times within a memeplex are 5; the maximum iteration times are 2000. For each case, 50 executions are performed to appraise the results. The simulation work is conducted on a laptop with a dual 2.4GHz CPU, 1G RAM and Window XP operating system. SFLA is coded in C program language and compiled in MS.Visual C++ 6.0 and the average computational time is 0.5 second. Table 2 shows the result.

Table 1. Demand for products in each period

Month	1	2	3	4	5
Demand	10	62	12	130	154
Month	6	7	8	9	10
Demand	129	88	52	124	160
Month	11	12			
Demand	238	41			

Table 2. Comparison result of SFLA with other heuristic algorithms

Algorithms	Cost
SFLA	501.2
WW	501.2
SM	501.2
EOQ	643.2
L4L	648
LUC	558.8

Synthetically considering the results of SFLA, it is feasible and effective for solving SLLS problem. With integrating more modifications and heuristics to SFLA, we believe the performance of SFLA can be improved a lot and it will become a more powerful tool to solve MLLS problem with series structure, assembly structure and general structure.

5. Conclusion and Future Research

In this paper, we applied a SFLA to solve uncapacitated SLLS problem. Our contribution is that we introduced SFLA to solve SLLS problem and gained ideal results. In our future study, we firstly should expend our experiment framework to more complex product structures with larger scale. Then more improvements should be made such as combining some heuristics with SFLA.

Acknowledgements

This research is supported by National Natural Science Foundation (China) under grant numbers 70971017, Natural Science Foundation of Zhejiang Province (China) under grant number (Y1100854), Philosophy and Social Science Foundation of Zhejiang Province (China) under grant number (10CGGL21YBQ), Youth Foundation of Humanities and Social Sciences of MOE (China) under grant number (12YJCZH065), Foundation of Zhejiang Educational Bureau (China) under grant number (Y201016979) and Science and Technology Research Program of Zhejiang Province (China) under grant number (2009C35007).

References

[1] Silver AE, Pyke F,Peterson F. Inventory Management and Production Planning and Scheduling. *3rd ed., New York: JohnWiley&Sons*; 1998.

[2] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks,* 1995; 1942-1948.

[3] Clerc M. Discrete particle swarm optimization illustrated by the traveling salesmanproblem. *http://www.mauriceclerc.net,*2000.

[4] Dellaert N and Jeunet J. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research*, 2000; 38: 1083-1099.

[5] Han Y, Kaku I, Zhou G, Cai J, Lin H. An application of PSO to uncapacitated multilevel lot-sizing problems with series structure. *Proceedings of the 2009 International Conference on Computational Intelligence and Security*, 2009; 2: 541-544.

[6] Tang O. Simulated annealing in lot sizing problems. *International Journal of Production Economics*, 2004; 88: 173-181.

[7] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. *Proceedings of 6th International Symposium on Micro Machine and Human Science*, 1995; 39-43.

[8] Shi Y, Eberhart R. Parameter selection in particle swarm optimization. *Proceedings of the 7th Annual Conference on Evolutionary Programming*, 1998; 591-600.

[9] Shi Y, Eberhart R, A modified particle swarm optimizer. *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998; 69-73.

[10] Yelle, LE. Materials requirements lot sizing: a multilevel approach. *International Journal of Production Research*, 1979; 17: 223-232.

[11] Dong Y, Tang JF, Xu BD, Wang DW. An application of Swarm Optimization to Nonlinear Programming, *Computers and Mathematics with Applications*, 2005; 49: 1655-1668.

[12] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J*ournal of Water Resources Planning and Management*, 2003; 129: 210-225.