Send Orders for Reprints to reprints@benthamscience.ae

A Hybrid Intelligent Optimization Algorithm for an Unconstrained Single Level Lot-sizing Problem

Yi Han[1,2,3,*], Wang Dezhi[1], Lifeng Mu[4], Jianhu Cai[1], Ikou Kaku[5] and Liping Zhao[6]

[1]*Shanghai University, Shanghai, China;* [2]*Zhejiang University of Technology, Hangzhou, China;* [3]*Huazhong University of Science and Technology, Wuhan, China;* [4]*Research Center for Technology Innovation and Enterprise Internationalization, Hangzhou, China;* [5]*Tokyo City University, Yokohama, Japan;* [6]*College of Information and Engineering, Jinhua Polytechnic, Jinhua, China*

Abstract: Scatter search algorithm (SSA) and shuffled frog leaping algorithm (SFLA) are two intelligent optimization algorithms. SSA was introduced to solve discrete optimization problems in 1977 and SFLA in 2003 was created for solving continuous optimization problems. Currently, These two algorithms had already been wildly applied to solving many engineering optimization problems. Within this paper, a hybrid algorithm, which combines SSA and SFLA, is presented in the hope that the hybrid algorithm can contribute a great deal to the advancement of intelligence optimization research. A test is done on an unconstrained single-level lot-sizing (SLLS) problem to further demonstrate the effectiveness and efficiency of this hybrid algorithm.

Keywords: Optimization, scatter search, shuffled frog leaping, single-level lot-sizing, unconstrained.

1. INTRODUCTION

Scatter search algorithm (SSA) is termed a kind of meta-heuristics along with genetic algorithm (GA) and taboo search (TS). Meta-heuristic, compared with heuristics, modifies and guides other heuristics to search for better results. So far, the past decades have seen the ability of SSA in solving many hard optimization problems [1]. Fred Glover invented the fundamental concepts and principles of SSA method and Manuel Laguna made extensive contributions afterwards [2]. SSA highly depends on decision combining rules and problem constraints when searching for a new solution in a vast solution space. Different from the other evolutionary methods such as GA and TS, scatter search is constructed based on the principle that through careful designs and clever methods for creating new solutions huge benefits might come. SSA employs strategies to deal with seeking diversification and intensification, which have been demonstrated highly appropriate in a large amount of engineering optimization problems. SSA is of great flexibility in that every components could be implemented by a great deal of ways and any degrees of complexity [3].

In 2003, shuffled frog leaping algorithm (SFLA) is created and used by Eusuff and Lansey [4]. SFLA is a group-based, novel and effective intelligent optimzation computing method. It's now receiving ever increasing interests from all walks of academic institutions and engineering optimization fields. SFLA, which is full of powerful optimum-reaching ability and can be implemented easily, combines the features of Memetic algorithm (MA) and particle swarm optimization (PSO). Therefore, it integrates strong local search (LS) ability and good global search (GS) c apability into itself [5-10].

Here, a hybrid algorithm combining charateristics of SSA and SFLA was provided hoping to bring both algorithms' advantages together. The effectiveness and efficiency was tested on an unconstrained SLLS problem. Computational result showed that the hybrid scheme functioned properly and satisfactorily.

2. BRIEF INTRODUCTION ON SSA

In SSA, there are several important parameters listed below [11].

- PSize = number of diverse solutions created through the *diversification generation method*
- b = content of the reference set (RefSet)
- b_1 = number of the high-quality solutions (subset) in RefSet
- b_2 = number of the solutions with high diversity (subset) in RefSet
- RefSet = an ordered pair composed of b_1 and b_2 like (b_1 b_2)
- P = the population

The basic executive process of SSA is as follows [12]:

- **Step 1:** Generate a starting set P and guarantee a high degree of diversity. Apply improvement strategies on P. Designate a set consists of several best vectors and several solution with diversification as the Refset.
- **Step 2:** Constitute some subsets according to subset generation rule and create new elements

1874-110X/14 2014 Bentham Open

- **Step 3:** Apply the improvement strategies used in Step 1 on the new solutions even though some of them maybe unfeasible.
- **Step 4:** Rebuild Refset. Repeat Steps 2, 3 and 4 until the Refset remains the same for several generations. Make some changes to candidates in Refset and reshape Refset. Stop when a specified limit is met.

With the advance of the executive procedure, 5 very important methods are employed and listed below:

- ***A diversification generation method*** is for creating diversified solutions.
- ***An improvement method*** is for transforming a candidate solution into a improved one (Neither the input nor the output is demanded to be feasible, even if the outcomes are expected to be so. If no improvement is acquired or even worse solution is gained, the input is kept as it used to be).
- ***A reference set update method*** is for building and maintaining a reference set, which consists of b solutions. Solutions, which have permissions to enter the reference set, depend on their qualities or their diversities.
- ***A subset generation method*** is for operating on the reference set to form 2-,3-,4-,5- and muti-element subsets.
- ***A solution combination method*** is for transforming a given subset of solutions into one or more candidate solutions.

The following steps illustrate the executive procedure of SSA [10].

- Start with $P=\emptyset$. Use the *diversification generation method* to construct a solution and apply the *improvement method*. Let x be the resulting solution. If $x \notin P$ then add x to P(*i.e.*, $P=P \cup x$), otherwise, discard x. repeat this step until $|P|=PSize$
- Use the *reference set update method* to build $RefSet = \{x^1,\ldots,x^b\}$ with the "best" b solutions in P. Order the solutions in *Refset* according to their objective function value such that x^1 is the best solution and x^b the worst. Set *NewSolutions* = TRUE
- While (*NewSolutions*) Do

  Gnerate *NewSolutions* with the *subset generation method*. Set *NewSolutions* = FALSE

  While (*NewSubsets*$\neq\emptyset$) Do

  Select the next subset *s* in *NewSubsets*.

  Apply the *solution combination method* to s to obtain one or more new trial solutions x. Apply the *improvement method* to the trial solutions

  Apply the *reference set update method*.

  If (*RefSet* has changed) Then

  set *NewSolutions* = TRUE

  End If

  Delete *s* from *NewSubsets*

  End While

  End While

3. BRIEF INTRODUCTION ON SFLA

SFLA is a newly popular intelligent optimization technology introduced in view of the memetic evolution in a group of frogs, which seek for the location where the maximum amount of available foods exist. SFLA, originated in 2003, combines the advantages of memetic algorithm and particle swarm optimization (PSO) technology. Within SFLA, the iterative population is formed with a group of frogs (candidate solutions) assigned into several subsets (memeplexes). In each memeplex, which is looked as a different culture (meme) of frogs, frogs carry out a local search. Here, songs, ideas, knowledge, fashion and ways of making products or ways of building houses can also be viewed as meme(s). A memetic evolution process goes on among frogs in each memeplex, *i.e.*, they extend local scout in solution space following one specific strategy or several ones which advocate the communication in a meme among local individuals. A certain number of memetic evolution later, information exchange process goes through among memeplexes during re-shuffling period. The local scout and the re-shuffling process alternate through out the whole process until a convergence criterion is reached [5-9].

The executive process of SFLA goes like: Constitute the initial population with P randomly generated candidate solutions (frogs). For problems with L-dimensions, a frog *can be presented with a vector such* as $X_i=(x_{i1}, x_{i2},\ldots, x_{iL})$. Then, order the frogs in a descending way according to each one's fitness or objective function. Afterwards, the whole population breaks into m subsets (memeplexes), each having n frogs ($P = m \times n$). During the dividing process, the first frog is put into the first memeplex, the second frog is assigned to the second memeplex and frog m enters the mth memeplex, while frog m+1 goes nowhere but the first memeplex, etc [10].

In every memeplex, the best frog and the worst one are labeled X_b and X_w. The best-of-all frog is termed X_g. Then, a process with similar formulas as in PSO functions to enhance X_w (not all the frogs within a memeplex) in each small subset. The formulas are as follows:

$$S = \text{rand}() \times (X_b - X_w) \quad (1)$$

$$X_{new} = X_w + S, -S_{max} \le S \le S_{max} \quad (2)$$

here *rand*() is a function with a returned float number, which is randomly generated between 0 and 1; S_{max} is the maximal allowed change times at a frog's position. If a better solution X_{new} is attained, the worst frog X_w goes to this position. Otherwise, formula (1) is adjusted by substituting X_g for X_b. If still no improvement happens in this situation, a newly created random solution is used to replace X_w. Therefore the memeplex is resorted and updated and afterwards all the memeplexes are re-shuffled together to facilitate exchanging information and reallocate frogs for the next search process. The flowchart of SFLA appears in Fig. (**1**) [4].

4. HYBRID ALGORITHM

In each generation in SSA, the *subset generation method* produces some subsets according to the reference set. Let's

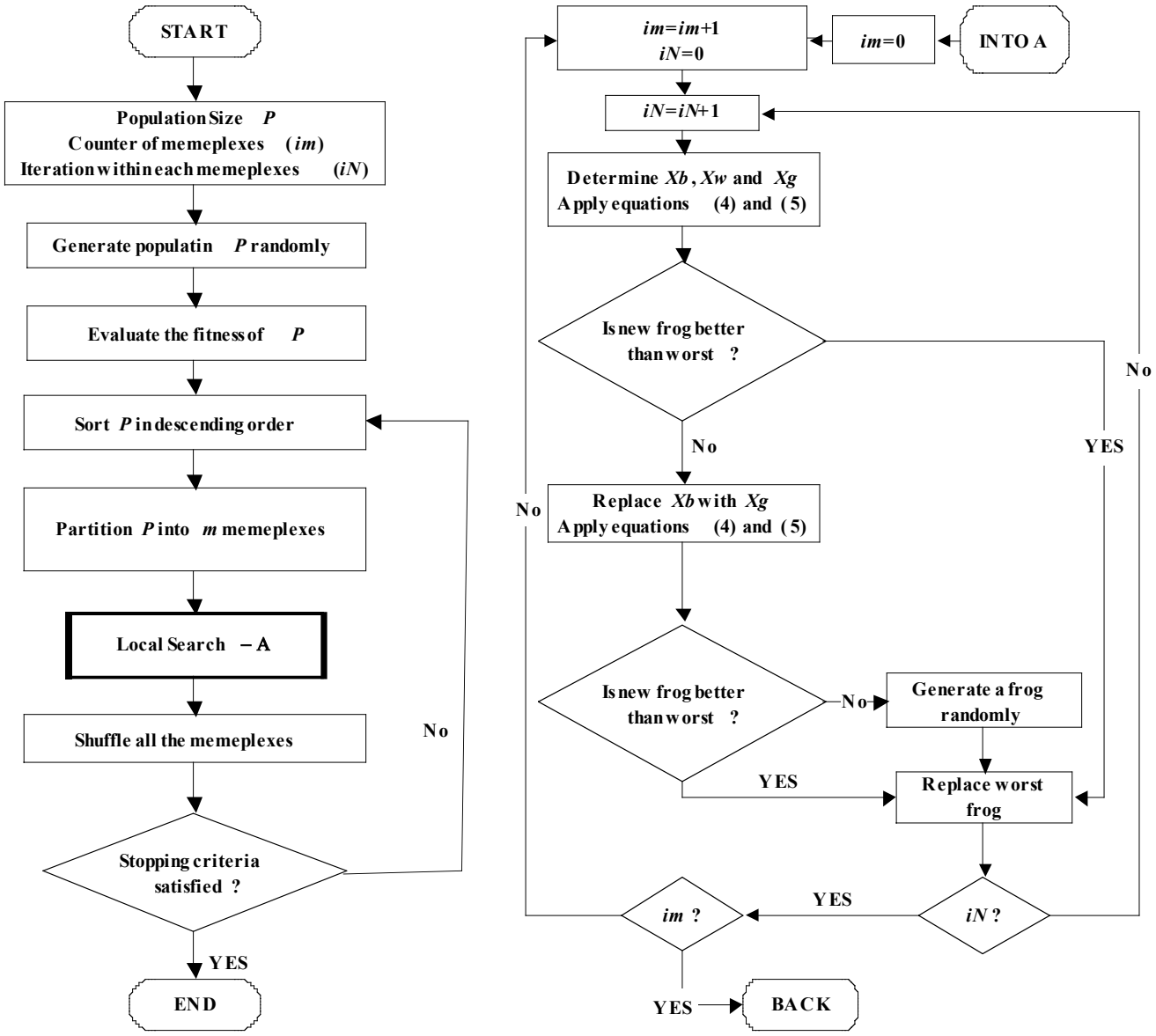

Fig. (1). Flowchart of SFLA.

say a reference set is of 3 satisfactory solutions and 2 good-diversity solutions and the index of each solution is among {1, 2, 3, 4, 5}. Then SSA generates firstly the 2-element subsets such as {{1,2}, {1,3}, {1,4}, {1,5}, {2,3}, {2,4}, {2,5}, {3,4}, {3,5}, {4,5}}; afterwards, 3-element subsets are generated by inserting another index, which indicates a solution with the the best diversity (largest distance) to each one in the current 2-element subset, into the 2-element subsets; therefore the 4-element subsets are set up in the same way by putting a designated solution into 3-element subsets; in the end, the reference set itself is used as a large subset.

Once all subsets are constituted, the *solution combination method* is applied to create a new solution. A classical way of combining a subset of solutions into a new solution is a scoring technology. Given that a 3-element subset $\{x_1(1,0,0,1), x_2(1,0,1,0), x_3(0,1,1,1)\}$ with $f(x_1)=5$, $f(x_2)=3$ and $f(x_3)=9$, the weight of each bit in each solution are {**0.294** (5/17), **0.176** (3/17), **0.53** (9/17)}. According to the weights of each bit in every solution, the weights of each bit in the new solution is {**0.47** (0.294*1+ 0.176 *1 + 0.53*0), **0.53** (0.294 *0+ 0.176 *0+ 0.53 *1), **0.706**(0.294 *0+ 0.176 *1+ 0.53*1), **0.824** (0.294*1 + 0.176*0+ 0.53*1)}. So, a new {0, 1, 1, 1} is decided since 0.47 is lower than 0.5 while other weights of other bits outgrow 0.5.

In Hybrid algorithm, 3-element, 4-element and multi-element subsets in SSA are taken as memeplexes. We can change the afore-mentioned best solution x_3 as $x_3^{'}$ = {0, 0.53, 0.53, 0.53} and the worst solution x_w as as $x_w^{'}$ {0.176, 0, 0.176, 0}. Then reshape (1) and (2) into (3) and (4).

$$S = 2 \times rand() \times (X_b - X_w) \quad (3)$$

$$X_{new} = X_w \pm S, -S_{max} \le S \le S_{max} \quad (4)$$

Here formula (3) and (4) are used to generate a new solution x_{new} (but the variables appear on the right side of (3) and (4) are $x_3^{'}$ and $x_w^{'}$). In (4), whether there will be a minus sign or a plus sign depends on the random number given by computer with a possibility of 0.5. If x_{new} is greater than 1, it is

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			

changed as $x_{new} = x_{new} - 1$ until x_{new} is smaller than 1; if x_{new} is smaller than 0, it is adjusted as $x_{new} = x_{new} + 1$ until x_{new} is a positive one. Here x_{new} is a real number vector. After comparing each bit in x_{new} with a number 0.5, x_{new} is then transferred into a binary vector.

The 2-element subsets can still experience a crossover operation as in GA to create new chromosomes. The other new solutions come out according to (3) and (4) as in SFLA. The logical executive steps of the hybrid algorithm are listed as:

- Initiate all the parameters in SSA
- Generate the Population with a certain number of solutions according to the *diversification generation method* and the *improvement method*
- Build up the reference set with 5 solutions selected from the population (3 best-quality ones and 2 good-diversity ones) according to the *reference set generation method*
- Produce 2-element, 3-element, 4-element and multi-element subsets according to the *subset generation method*
- Apply different *solution generation method* to different types of subsets (crossover operation for 2-element subsets and SFLA's new solution generation method for the other types of subsets)
- Sequencing the newly produced solutions together with the solutions not used before to generate reference set to form a new population with the same size as before
- IF the stopping limits are met, return the first solution in the ordered population; else go to Step 3.

5. EXPERIMENTAL RESULTS

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.

Here the proposed hybrid algorithm is programmed and implemented in C++ language with a laptop having a dual 2.4GHz CPU, 1G RAM and Window XP operating system, compiled in MS. Visual C++ 6.0 environment and tested with an uncapacitated SLLS problem.

In SLLS problem, there is only one product under planning with T periods (in this paper, it is set to be 12). The SLLS problem can be coded 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. The objective fuction is composed by inventory volume ×inventory keeping cost + setup cost × production times (if there is a '1' in a solution, the objective function is increase by a setup cost).

In this paper, an unconstrained SLLS case having 1 product and 12 production periods is listed in Table **1**. The setup cost is 54 and the inventory keeping cost is 0.4. The Population size is 20, the size of reference set is 5, the max iterative times is set as 200 and the inner iteration times in a meme group is 2. The hybrid algorithm is run 50 distinctive times. The computational result shows that the algorithm performs each run with an average CPU time of 0.6 second. In each run, hybrid algorithm attained the optimal cost 501.2. The result of the hybrid algorithm is compared with results of some famous ever-existed heuristic algorithms such as Wagner-Whitin (WW), Silver-Meal (SM), EOQ, Lot-4-Lot and Least Unit Cost (LUC) method as listed in Table **2**.

Table 2. Comparison between hybrid algorithm and other heuristic algorithms.

Algorithms	Cost
Hybrid Algorithm	501.2
WW	501.2
SM	501.2
EOQ	643.2
L4L	648
LUC	558.8

Through synthetically analysis on the result presented by the new algorithm, we can conclude that the proposed algorithm is highly feasible, effective and efficient to solve SLLS problem with no capacity constains. Through integrating more modifications, improvements and heuristics to the new algorithm, it is reasonable to assert that the performance of which can be improved a great deal and that it is of foreseeable great potential to advance towards a powerful tool suitable oin many other optimization occasions.

CONCLUSION

This paper presents a hybrid algorithm combining the SSA and SFLA together. Through a test for its effectiveness and efficiency on an unconstrained SLLS problem, the proposed scheme shows satisfactory performance and strong potential to develop into another powerful intelligent optimization tool suitable for solving many complicated and sophisticated optimization problems in many academic and engineering fields.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported in part by grants from NSFC under grant No.71301147, No.71301148, No.71302051, the Humanities and Social Science Project of Ministry of Education of China under grant No.12YJCZH065, Japan Society of Promotion of Science (JSPS) under grant No.21510148, and Foundation of Zhejiang Education Committee (No. Y201225266), the Science and Technology Project of Jinhua (No.2012-3-060).

REFERENCES

[1] R. Marti, "Scatter search-wellsprings and challenges," *European Journal of Operational Research*, vol. 169, pp. 351-358, February 2006.

[2] R. Marti, M. Laguna and F. Glover, "Principles of scatter search," *European Journal of Operational Research*, vol. 169, pp. 359-372, February, 2006.

[3] M. Laguna, "Global optimization and meta-heuristics," *www.mat.univie.ac.at/~neum/glopt/mss*, 2004.

[4] M. Eusuff and K. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, pp. 210-225, March, 2003.

[5] M. Eusuff, K. Lansey and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, pp. 129-154, February, 2006.

[6] A. Rahimi-vahed and A. Mirzaei, "A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem," *Computers and Industrial Engineering*, vol. 53, pp. 642-666, January, 2007.

[7] A. Rahimi-vahed, M. Dangchi, H. Rafiei and E. Salimi, "A novel hybrid multi-objective shuffled frog-leaping algorithm for a bicriteria permutation flow shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 41, pp. 1227–1239, November, 2009.

[8] B. Amiri, M. Fathian and A. Maroosi, "Application of Shuffled frog-leaping algorithm on clustering," *International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 199-209, November, 2009.

[9] H. Elbehairy, E. Elbeltagi, T. Hegazy and K. Soudki, "Comparison of two evolutionary algorithms for optimization of bridge deck repairs," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, pp. 561-572, September, 2006.

[10] E. Elbeltagi, T. Hezagy and D. Grierson, "Comparison among five evolutionary -based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 43-53, January, 2005.

[11] A. Haq, M. Saravanan, A. Vivekjaj and T. Prasad, "A scatter search approach for general flowshop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 31, pp. 731-736, January, 2007.

[12] F. Glover, M. Laguna and R. Marti, "Fundamentals of scatter search and path relinking," *Control and Cybernetics*, vol. 39, pp. 653-684, March, 2000.

Received: September 22, 2014 Revised: November 30, 2014 Accepted: December 02, 2014