Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (14): 2752-2759, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2752.2759

A Scheme for Combining SSA with HSA and its Application to an Uncapacitated SLLS Problem

[1,2]Yi Han, [1]Jianhu Cai, [3]Ikou Kaku, [4]Yanlai Li and [5]Liping Zhao
[1]College of Economics and Management,
[2]Research Center for Technology Innovation and Enterprise Internationalization,
Zhejiang University of Technology, Hangzhou, 310023, China
[3]Department of Management Science and Engineering,
Akita Prefectural University, Yulihonjo, 015-0055, Japan
[4]School of Traffic, Transportation and Logistics,
Southwest Jiaotong University, Chengdu, 610031, China
[5]College of Information and Engineering, Jinhua Polytechnic, Jinhua, 321017, China

Abstract: Scatter Search Algorithm (SSA) and Harmony Search Algorithm (HS) are two meta-heuristics proposed respectively in 1977 to solve discrete optimization problems and in 2001 for continuous optimization problems. Currently, both algorithms have been wildly adopted in many optimization fields. In this study, a scheme for combining the scatter search algorithm and harmony search algorithm is proposed with a hope that the proposed algorithm can play an important role in the developing process of intelligence optimization and be suitable for many other research fields. The effectiveness and efficiency are tested through an uncapacitated Single-level Lot-sizing (SLLS) problem. Computational result shows the feasibility of the proposed scheme. The significance of this study is proposing a new algorithm for discrete optimization field and a new tool for SLLS problem.

Key words: Scatter search algorithm, harmony search algorithm, uncapacitated, SLLS, meta-heuristics

INTRODUCTION

Scatter Search Algorithm (SSA) along with Genetic Algorithm (GA) and Tabu Search (TS) is one of meta-heuristics. Meta-heuristic is a main mechanism that modifies and guides other heuristics to produce better results than those of heuristics. SSA has been successfully applied to solve many hard optimization problems (Marti, 2006; Laguna and Marti, 2003). The fundamental concepts and principles of the method were first introduced in 1977 by Fred Glover and extensive contributions have been made by Laguna (Glover, 1977; Marti *et al*., 2006). SSA is based on combining decision rules and problem constraints. In contrast to other evolutionary methods such as GA, scatter search is founded on the premise that systematic designs and methods for creating new solutions afford significant benefits. The scatter search methodology is very flexible since each of its elements can be implemented in a variety of ways and degrees of sophistication (Laguna, 2002; Geem *et al*., 2001).

Harmony search algorithm (HS) is a novel and global-search based intelligent optimization algorithm which is proposed by Geem *et al.* (2001). HS mimics the improvisation process of musicians by constantly adjust the pitches of each instrument during the concert. So far, it has been comprehensively focused on and applied in many optimization research fields.

Consider musical instruments of five musicians on the Jazz bandstand as follows: Guitarist, Trumpeter, Drummer, Saxophonist and Double bassist. There are sets of preferable pitches in their memory, that is Guitarist: {Do, Mi, Sol}; Trumpeter: {La, Si, Sol}; Drummer: {Re, Sol, Si}; Saxophonist: {Fa, Do, La}; Double bassist: {Re, Sol, Mi}. During the concert, Guitarist randomly improvises {Do} from his memory, Trumpeter improvises {Sol} from his memory, Drummer adjusts {Re} from his memory to come up with {Fa}, Saxophonist improvises {La} from his memory and Double bassist improvises {Si} from the available range {Do, Re, Mi, Fa, Sol, Si}. All these pitches together form a fresh harmony {Do, Sol, Fa, La, Si} as shown in Fig. 1 and 2 (Al-Betar and Khader, 2012).

As a new novel optimization method, HS is of simple principle and easy to be implemented. It doesn't matter if the objective function is differentiable or derivable. HS can be applied to continuous optimization problems and discrete ones. HS which showed its superiority over Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) and so on in many optimization fields,

Corresponding Author: Jianhu Cai, College of Economics and Management, Zhejiang University of Technology, Hangzhou, 310023, China Tel: +86-15669034079

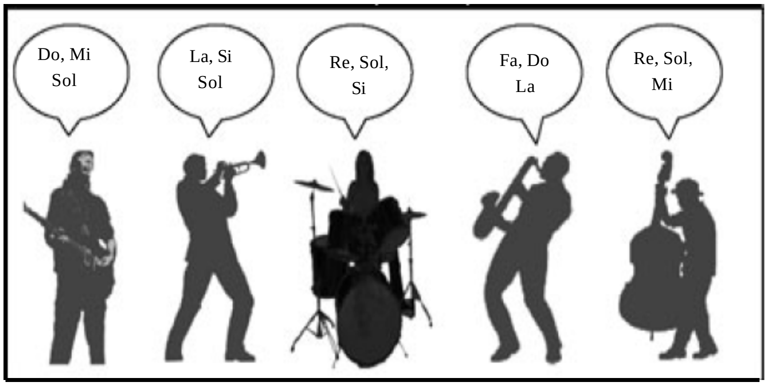

Fig. 1: Improvision process of musicians

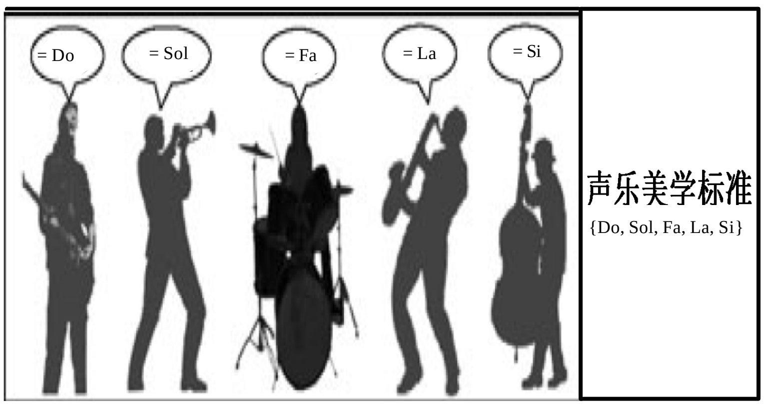

Fig. 2: Result of improvision

provides a new idea and novel method for solving combinatorial optimization, water conservancy and civil engineering and parameters setting optimization, etc.

In this study, a hybrid algorithm of HS and SSA is proposed. The scheme and executive procedure are depicted in detail firstly. Then the hybrid algorithm is applied to SLLS problem to check its ability. The computational result shows the hybrid algorithm is suitable for small scale binary discrete optimization problems.

UNCAPACITATED SLLS PROBLEM

Single level lot-sizing problem is an important decision making problem in industry. Single level means there is only one product is involved. All what we need to do is to give out a production plan without considering the availability of resources within a planning span.

Here, we suppose that the planning span is limited to T, the demands d_t for each period in the span are known before. The production cost and inventory keeping fee will not change with the number of production volume and planning period T. The lead time is 0 and no backlogging is allowed. The setup cost for each lot is a constant and the inventory level in the starting period and the last period is 0.

The mathematical model goes like:

Minimize:

$$Z = \sum_{t=1}^{T}(h_t \times I_t + s_t \times Y_t) \quad (1)$$

Subject to:

$$I_{t-1}+X_t-I_t = d_t \;\; t = 1 \cdots T \quad (2)$$

$$X_t \leq MY_t \quad t = 1 \cdots T \quad (3)$$

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

$$X_t \geq 0, I_t \geq 0 \quad t = 1 \cdots T \quad (5)$$

$$I_0 = I_t = 1 \quad (6)$$

where, M is a very large number; h_t is the inventory keeping fee per unit at period t; s_t is the setup cost per unit at period t; X_t is the production volume at period t and Y_t is the binary decision variable indicating if the production activity is carried on.

The objective function (1) Aims at minimizing the sum of setup cost and inventory cost throughout the planning span. Constraint (2) Is the logistics balancing formula. Constraint (3) Shows you can produce as much as needed if there is a production activity indication. Constraint (4) means Y_t is a binary indication variable. Constraint (5) claims production volume is 0 at least and there is no stockout in the warehouse. Constraint (7) requires the starting inventory level and the ending inventory level are 0.

BRIEF INTRODUCTION TO SSA

Before we present the basic SS approach, the following parameters should be listed out (Al-Betar and Khader, 2012):

P	=	Size of the set of diverse solutions generated by the diversification generation method
b	=	Size of the reference set (RefSet)
b	=	size of the high-quality subset of RefSet
b	=	size of the diverse subset of RefSet
b	=	the population where the RefSet is formed

The basic SS approach is sketched as follows (Glover *et al.*, 2000):

- Generate a starting set of solution vectors P to guarantee a critical level of diversity and apply heuristic processes designed for the problem as an attempt for improving these solutions. Designate a subset of the best vectors to be reference solutions. The notion of "best" in this step is not limited to a measure given exclusively by the evaluation of the objective function. In particular, a solution may be added to the reference set if the diversity of the set improves even when the objective value of the solution is inferior to other solutions competing for admission into the reference set
- Create new solutions consisting of structured combinations of subsets of the current reference solutions. The structured combinations are:
    - Chosen to produce points both inside and outside the convex regions spanned by the reference solutions
    - Modified to yield acceptable solutions
- Apply the heuristic processes used in Step 1 to improve the solutions created in Step 2. These heuristic processes must be able to operate on infeasible solutions and may or may not yield feasible solutions
- Extract a collection of the "best" improved solutions from Step 3 and add them to the reference set. The notion of "best" is once again broad; making the objective value one among several criteria for evaluating the merit of newly created points. Repeat Steps 2, 3 and 4 until the reference set does not change. Diversify the reference set, by re-starting from Step 1. Stop when reaching a specified iteration limit

During the execution of the aforementioned procedures, 5 very important methods which are key factors should be adopted (Amiri *et al.*, 2009):

- A diversification generation method to generate a collection of diverse trial solutions, using an arbitrary trial solution (or seed solution) as an input
- An improvement method to transform a trial solution into one or more enhanced trial solutions (Neither the input nor the output solutions are required to be feasible, though the output solutions will more usually be expected to be so. If no improvement of the input trial solution results, the "enhanced" solution is considered to be the same as the input solution)

- A reference set update method to build and maintain a reference set consisting of the b "best" solutions found (where the value of b is typically small, e.g., no more than 20), organized to provide efficient accessing by other parts of the method. Solutions gain membership to the reference set according to their quality or their diversity
- A subset generation method to operate on the reference set, to produce a subset of its solutions as a basis for creating combined solutions
- A solution combination method to transform a given subset of solutions produced by the subset generation method into one or more combined solution vectors

Figure 3 shows the interaction among these five methods (Glover *et al.*, 2000).

BRIEF INTRODUCTION ON HS

Improvision, harmony and musician: In a big concert, the improvisation is played by many different musical instrument musicians such as drumers, guitar players, saxphone players, pianist and so forth to present a harmonious rhythm. In HS, the improvisation equals to the population of solutions, the harmony corresponds to each candidate solution and the musician is referred to as each decision variable or bit in the candidate solution.

Notes and Pitch range: Notes which represent the bits in each solution, are the melody played by the musical instruments. Pitch range which corresponds to the domain of each bits, is the changing range for notes. Figure 4 shows the mapping relationship between notes and bits.

- Start with P=Ø. Use the **diversification generation method** to construct a solution and apply the improvement method. Let x be the resulting solution. If **$x \subsetneq P$** then add x to P(i.e., P = PUx),otherwise,discard x. repeat this step until |P| = PSize
- Use the **reference set update method** to build RefSet = {x^1,…,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. Make **NewSolutions** = TRUE.
  while (NewSolutions) **do**
- Gnerate **NewSolutions** with the subset generation method. Make **NewSolutions** = FALSE.
  while (**NewSubsets**≠Ø) 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**
    - Make **NewSolutions** = TRUE
    end if
  - Delete s from **NewSolutions**
    end while
    end while

Fig. 3: Basic executive procedure of SSA

Audio-aesthetic standard, harmony memory and pleasing harmony: In HSA, audio-aesthetic standard equals to the objective function; harmony memory is a set of all candidate solutions; and pleasing harmony corresponds to the best solution. Figure 5 shows the corresponding relationship between audio-aesthetic standard and the objective function.

Parameters: Like the other intelligent algorithms, HSA contains some important parameters. There are Harmony Memory Size (HMS) which indicates the size of the harmony memory, Harmony Memory Consideration Rate (HMCR) which is the probability of choosing a variable in a candidate solution, iteration number (N) which limits the total running times, Pitch Adjustment Rate (PAR) which is the probability of adjusting a bit in a newly generated solution and Band Width (BW or §) which represents the biggest step for changing a certain bit in a newly generated solution.

Three operators: When musicians play a note with their own instruments, each one has three choices: (1) Choosing a note from all harmonies in his memory, (2) Modify and adjust a note in a chosen harmony from his memory to create a new harmony, (3) Randomly choosing a note from the pitch range to substitute a note in a chosen harmony to form a new harmony. During the solving process of optimizing a problem, each new solution is generated according to the following three ways: (1) Choosing a candidate solution from the population, (2) Modify a bit in a chosen candidate

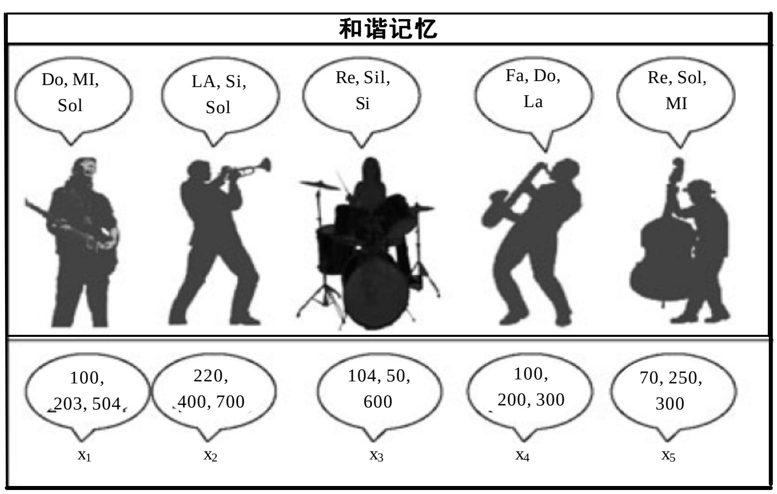

Fig. 4: Harmony memory

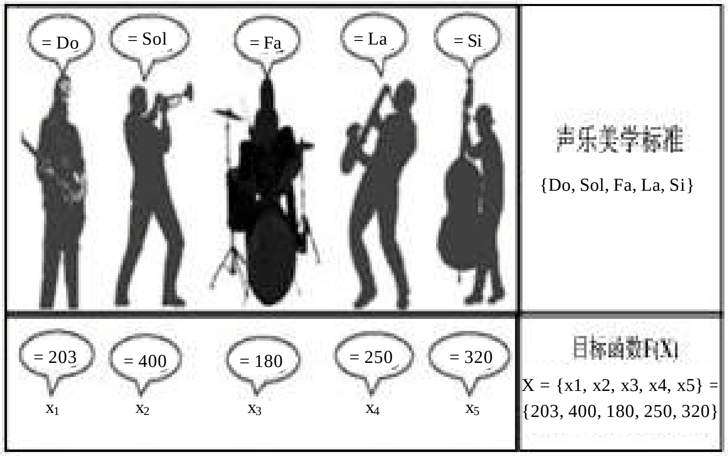

Fig. 5: Candidate solution and it's objective function

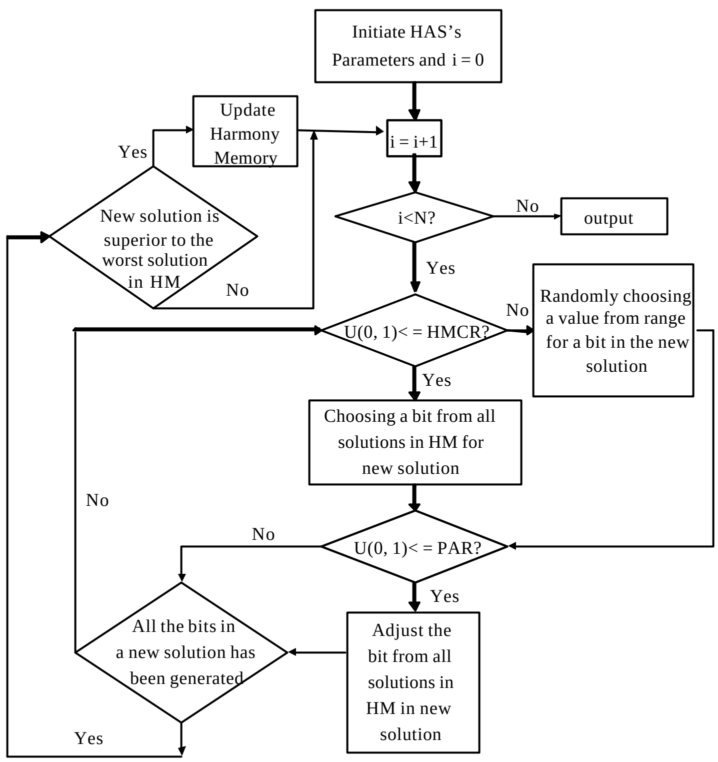

Fig. 6: Flowchart of HSA

solution to create a new solution; (3) Randomly selecting a bit from the range to replace a bit in a chosen candidate solution to form a new solution.

In 2001, Geem *et al*. (2001) related the aforementioned three generating-harmony choices with three new-solution generating ways to form memory selection, variable ajustment and random selection operators (Geem *et al*., 2001). These operators are excecuted according to HMCR and PAR.

Ceasing criteria and flowchart: Like the other intelligent algorithms, HSA has many ceasing criteria. The usually adopted two measures are: (1) In the latest several times, there is no apparent improvement on the harmony memory, (2) The presetted iteration times are met. Two measures can be both adopted or respectively adopted during the process of HSA. Figure 6 shows the flowchart of HSA.

A SCHEME OF HYBRID SSA (SSA-HSA)

Coding and Decoding: Since we adopted SLLS problem to test the newly proposed algorithm, the coding is like a string of binary variables as shown in Fig. 7. The decoding is performed according to the bianry string. If there is a 1 in a bit, the production activity starts with the production volume equaling to all the demands between two periods having 1. Figure 8 shows the decoding result from Fig. 7 if the demand in each period is 10.

Crossover and mutation operator: In SSA-HSA, crossover and mutation opertors in GA are applied to 2-element subset to produce two new candidate solutions. Figure 9 and 10 show the crossover and muation operators.

SSA-HSA: In SSA, the subset generation method generates some subset from reference set. Suppose a reference set is of 3 satisfactory solutions and 2 good-diversity solutions, i.e., the indices of each solution are listed as {1, 2, 3, 4, 5}. The way for generating subsets is to firstly produce the 2-element subset like {{1,2}, {1,3}, {1,4}, {1,5}, {2,3}, {2,4}, {2,5}, {3,4}, {3,5}, {4,5}}; then 3-element subset is generated by insert another solution index which indicate the solution with the 'largest' distance to those solutions in 2-element subset, into 2-element subset; afterwards 4-element subsets are built up by inserting a 'largest' distance solution into 3-element subsets; finally the whole reference set is taken as a subset.

After all the subsets are formed, the solution combination method is applied to produce a new solution. The traditional way to combine a subset into a new solution is a scoring method. Suppose a 3-element subset {x_1(1,0,0,1), x_2(1,0,1,0), x_3(0,1,1,1)} with f(x1) = 5, f(x2) = 3 and f(x3) = 9, the weights of each solution are {0.294 (5/17), 0.176 (3/17), 0.53 (9/17)}. According the weights of each 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, the new solution is decided according to the weights as {0, 1, 1, 1} since 0.47 is lower than 0.5 while other weights are larger than 0.5.

In the proposed scheme, the reference is constructed with 3 best candidate solutions and 2 good-diversity candidate solutions (best and good-diversity solutions are decided according to the objective function). We adopted crossover and mutation operators as in GA for 2-element subset. The 3-element, 4-element and multi-element subsets in SSA are treated as harmony memory. When producing a new solution, a harmony search principle is adopted. The algorithm is stopped if the best solution is not changing apparently or if the reference remains the same as in last iteration or the maximum iteration times is reached. The specific executive procedure is depicted in Fig. 11 and 12.

Ceasing criteria: Here, we adopted the maximum iteration times which is set to be 8000, as the stopping rule.

EXPERIMENTAL RESULTS

The ucapacitated single level lot-sizing problem which aims at minimizing the setup cost, the production cost, the inventory carrying cost, the back ordering cost and so on, is the key decision making problem in materials requirements planning (MRP) system. Here the proposed SSA-HSA is programmed and implemented in C^{++} language and tested with uncapacitated SLLS problem.

Fig. 7: Coding of SLLS problem

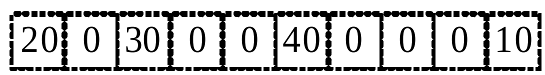

Fig. 8: Coding of SLLS problem

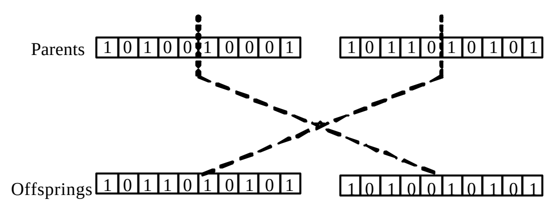

Fig. 9: Crossover operator

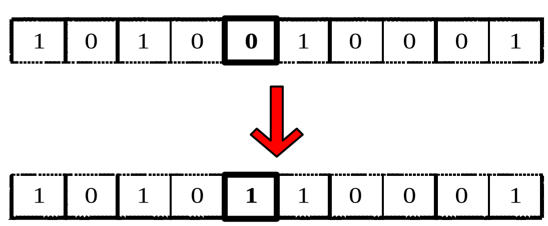

Fig. 10: Mutation operator

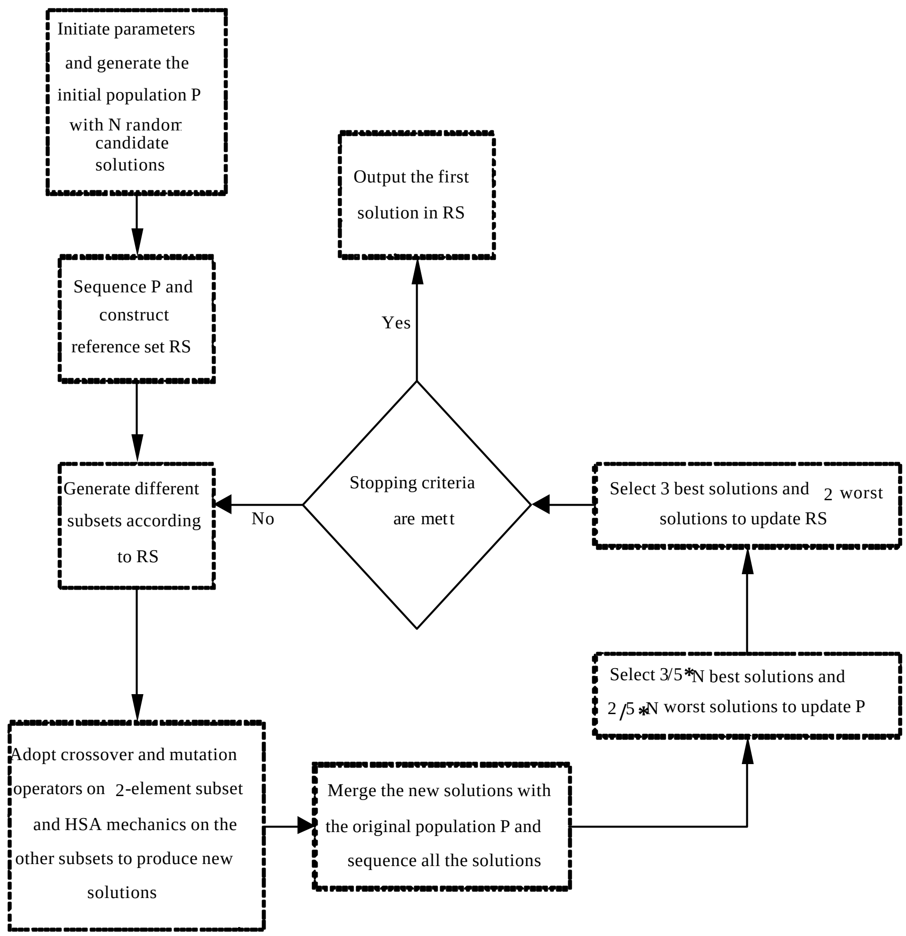

Fig. 11: Flowchart of SSA-HSA

Step 1: Initiate the parameters in **SSA-HSA**
Step 2: Generate the **Population** with a certain number of solutions according to **diversification generation method** and **improvement method**
Step 3: Build up the reference set with normally 5 solutions selected from the Population (3 satisfactory ones and 2 good-diversity ones) according to **reference set generation method**
Step 4: Generate **2-element** subsets, **3-element** subsets, **4-element** subsets and **multi-element** subset according to subset generation method
Step 5: Perform different **solution generation method** to different types of **subsets** (crossover operation for 2-element subsets and HSA's new solution generation method for the other types of subsets)
Step 6: Sequencing the newly generated solutions with the rest solutions in **Population** to form a new **Population** with the same size as before.
Step 7: If the stopping criteria are met, output the **first solution** in the current **Population**. Otherwise, go to Step 3

Fig. 12: Executive steps of SSA-HSA

Table 1: Demands for product 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			

An uncapacitated SLLS example with 1 product and 12 periods is shown in Table 1. The setup cost is 54 and the inventory keeping cost is 0.4. The number of Population in hybrid algorithm is 30, the reference set size is 5, the max iteration is 8000. The proposed SSA-HSA is executed 50 times respectively with an average CPU time of 2.15 sec. The result is listed in Table 2 along with those of some heuristic algorithms such as Wagner-Whitin (WW), Silver-Meal (SM), EOQ, Lot-4-Lot and Least Unit Cost (LUC).

According to the result, we can see the proposed SSA-HSA is feasible and effective for solving SLLS

Table 2: Comparison result for several heuristic algorithms

Algorithms	Best	Worst	Percent of best (%)	Average time (sec)
SSA-HSA	501.2	511.2	60	2
WW	501.2	/	100	/
SM	501.2	/	100	/
EOQ	643.2	/	0	/
L4L	648.0	/	0	/
LUC	558.8	/	0	/

problem. By Integrating more modifications and heuristics to the proposed algorithm, we believe that the performance of which can be improved a lot and that it is of great potential to be a powerful tool applicable to many other scientific research fields.

CONCLUSION

This study proposed a scheme for combining the SSA and HSA together. After a test of its effectiveness and efficiency through an uncapacitated SLLS problem, the hybrid algorithm showed acceptable performance and potential of being listed as another powerful tool suitable for solving many optimization problems in many research fields.

ACKNOWLEDGMENTS

This study was supported by NSFC under Grants No. 71202140, 71301147, 71302051 and 71301148, the H and S Science Project of MOE under Grant No. 10YZC630009 and 12YJCZH065, the Social Science Foundation Project of 'Zhijiang Youth' of Zhejiang Province under Grant No. 11ZJQN064YB, the Chinese Post-Doctorial Science Foundation under Grants No. 2012T50651, 2011M500858 and 2012M511620 and Zhejiang Provincial Key Research Insitute of Philosophy and Soical Science under Grant No. RWSKZD03-201201 and partially supported by JSPS under Grant No. 21510148.

REFERENCES

Al-Betar, M.A. and A.T. Khader, 2012. A harmony search algorithm for university course timetabling. Ann. Oper. Res., 194: 3-31.

Amiri, B., M. Fathian and A. Maroosi, 2009. Application of shuffled frog-leaping algorithm on clustering. Int. J. Adv. Manuf. Technol., 45: 199-209.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.

Glover, F., 1977. Heuristics for integer programming using surrogate constraints. Decis. Sci., 8: 156-166.

Glover, F., M. Laguna and R. Marti, 2000. Fundamentals of scatter search and path relinking. Control Cybern., 29: 653-684.

Laguna, M. and R. Marti, 2003. Scatter Search: Methodology and Implementations in C. 1st Edn., Lower Academic Publishers, Boston, Pages: 291.

Laguna, M., 2002. Global Optimization and Meta-Heuristics. In: Optimization and Operations Research, Derigs, U. (Ed.). Eolss Publishers, Oxford, UK.

Marti, R., 2006. Scatter search-wellsprings and challenges. Eur. J. Operat. Res., 169: 351-358.

Marti, R., M. Laguna and F. Glover, 2006. Principles of scatter search. Eur. J. Operat. Res., 169: 359-372.