Local Search for Sequencing of Storage and Retrieval Requests in Multi-Shuttle Automated Storage and Retrieval Systems

Xuan-Thuong Tran, Thanh-Do Tran, and Hwan-Seong Kim

***Abstract*—The sequencing of storage and retrieval requests in multi-shuttle automated storage and retrieval systems is formulated based on a new approach—the complete grouping formulation. The main objective of this sequencing problem is to minimize total travel time of a storage and retrieval machine to complete a list of storage and retrieval requests. Different local search algorithms are employed to solve the formulated sequencing problem. Since no local search suitable for the formulated problem is currently available, we have proposed several variation operators integrated into common local search algorithms for solving the problem. The performance of the proposed algorithms are examined and evaluated through extensive computational experiments. Through the experiments, suitable local search as well as effective variation operator are suggested.**

***Index Terms*—scheduling, sequencing problem, local search, grouping problem, automated storage and retrieval system.**

I. Introduction

NOWADAYS, automated storage and retrieval systems (AR/RS) have been used widely in industry from production systems to warehousing and distribution centers. In [1], the authors provided an overview of most important issues relating to the design, control, and performance measurement of AS/RS systems. Efficient control of AS/RS systems contributes to increasing the system throughput which is one of the most important objectives of an AS/RS system. This can be obtained through various control policies as reviewed in [1], including storage assignment, batching, parking of idle storage machines, and sequencing.

The sequencing of storage and retrieval requests has attracted many researchers [2], [3], [4]. For the class of the basic sequencing problem (i.e. for unit-load AS/RS systems), in [2], the authors dealt with a sequence of dual-cycle problem that minimizes total travel time of the R/S machine. The paper used an algorithm that combines the Hungarian method and the ranking algorithm for the assignment problem with tour-checking and tour-breaking algorithms. In [3], [4], these authors considered the sequencing problem with dedicated storage and block sequencing approach. They presented several optimum and heuristic methods for solving the sequencing problem. For the extensions of the basic sequencing problem such as sequencing problem with due time, twin-shuttle AS/RS, or multiple input/output locations per aisle were reviewed in [1].

At mentioned in [1], only a few works have been introduced for AS/RS systems with multi-capacity R/S machine that can operate in multi-cycle sequencing modes (quadruple or more). In this paper, we will present the sequencing problem of storage and retrieval requests in multi-shuttle AS/RS systems. The sequencing problem (SP) is formulated and solved in a general case with multi-capacity R/S machine by decomposing the problem into two subproblems. The first subproblem called complete grouping problem which finds an optimal grouping way, i.e. the optimal combination of the storage and retrieval requests. The second subproblem is routing problem which finds an optimal route for R/S machine to visit all locations in each group.

The sequencing of storage and retrieval requests is NP-hard problem in general [5], even some special cases can be solved in polynomial time [4]. Thus, for this class of problem, (meta-) heuristics are commonly employed to approximate the optimal solution. In this work, we apply simple local search algorithms (stochastic hill climbing, random-restart hill climbing) and iterated local search to solve the formulated SP. Since no operator is available in literature, we propose some operators work with local search algorithms and evaluate their performances through computational experiments. This work aims to obtain two goals: the first goal is to find suitable operators for the sequencing of storage and retrieval requests in multi-shuttle AS/RS system; the second one is to compare and evaluate the performances of local search algorithms and iterated local search.

II. Problem Description and Formulation

A. Problem Description

A multi-shuttle AS/RS system is considered in this paper. The system consists of a multi-shuttle crane working in one aisle with two storage racks and one input/output location. A stored unit-load (pallet/tote, . . .) enters the input location on a conveyor and waits for R/S machine to be picked and moved to the designate storage location. Commonly, these stored loads are performed according to First Come First Serve (FCFS) rule. On retrieval process, the R/S machine moves to the location of retrieved load, pick it up and then moves to output location. In single-shuttle unit-load AS/RS, the R/S machine can perform maximum one storage and one retrieval request in one cycle. Meanwhile, in multi-shuttle AS/RS

Manuscript received January 08, 2014; revised January 19, 2014. This work was supported by a grant (11 Transportation System – Logistics 02) from the Transportation System Efficiency Program funded by the Ministry of Land, Infrastructure and Transport of the Korean Government. TDT is supported by the CORDI-S Doctoral Fellowship of the French National Institute for Research in Computer Science and Control.

Xuan-Thuong Tran is with the Graduate School and Department of Logistics, Korea Maritime and Ocean University, Busan, South Korea; e-mail: xuanthuong@kmou.ac.kr.

Thanh-Do Tran is with Inria Lille – Nord Europe and Université Lille 1, Villeneuve d'Ascq, France; e-mail: thanh-do.tran@inria.fr.

Hwan-Seong Kim is with the Department of Logistics, Korea Maritime and Ocean University, Busan, South Korea; e-mail: kimhs@kmou.ac.kr.

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

system, R/S machine can perform more than two requests in one cycle, e.g two storage and two retrieval requests in dual-shuttle AS/RS and up to six requests in triple-shuttle AS/RS. Therefore the system throughput can be increased.

Since the lists of retrieval requests are changed overtime, sequencing of retrieval requests need to be treated as dynamic problem. According to [5], there are two sequencing methods to deal with this issue: block sequencing and dynamic sequencing. In this paper, we consider block sequencing method with dedicated storage strategy. Therefore the lists of storage and retrieval requests are known beforehand and no new requests are added until one block is finished.

Due to the higher capacity of R/S machine, R/S machine can visit more locations in one cycle results in more available routes for R/S machine to finish a cycle. As consequence, the sequencing of storage and retrieval requests consists of two tasks. The first task is to find an optimal combination of storage and retrieval requests, called grouping problems. The second one is to find an optimal route inside each group so that the R/S machine can visit all storage and retrieval locations in one cycle. The detail formulation of the two subproblems will be presented in Sect. II-C.

B. Travel Time Calculation

The main objective of sequencing problem in this paper is to minimize the total travel times of the R/S machine. Therefore the cost of a solution is considered as the total travel times of R/S machine. Various travel time models for AS/RS were proposed in literatures [6], [7]. For simplicity, in this paper we will use the Tchebyshev metric to compute the travel times of R/S machine.

The travel time between two arbitrary locations l_i and l_j, and between buffer (or I/O) location and arbitrary location l_i are calculated as in Eq. 1 and Eq. 2, respectively.

$$t_{ij} = \max\left(\frac{|x_i - x_j|}{hs}, \frac{|y_i - y_j|}{vs}\right) \quad (1)$$

$$t_{bi} = t_{ib} = \max\left(\frac{|x_i|}{hs}, \frac{|y_i|}{vs}\right) \quad (2)$$

where,
hs, vs: are the horizontal speed and vertical speed of R/S machine, respectively.
(x_i, y_i): is the coordinate of rack location i.

It is noted that in this paper the travel times from I/O station to all rack locations and between two arbitrary rack locations were calculated beforehand. This will help to improve computational speed during searching processes.

C. Problem Formulation

The sequencing problem of storage and retrieval requests was decomposed into two subproblems which are formulated in this section.

1) Formulation of Complete Grouping Problem: Various well-known combinatorial optimization problems that can be considered as original grouping problems such as cell formation problem, various timetabling problems, graph coloring problem, bin packing problem, multiple traveling salesman problem, multiple knapsack problem, etc [8]. However, in original grouping problems the number of groups can be changed and number of items in each groups can equal or different between the groups. While in our sequencing problem the number of group is fixed and known, and each group contains exact k storage and k retrieval requests, where k is the capacity of R/S machine. In order to facilitate the formulation, some assumptions and notations are needed.

Assumptions:

- The numbers of storage requests and retrieval requests in each block are equal and known beforehand.
- The R/S machine starts and ends a trip at the buffer zone (I/O).
- The number of storage/retrieval requests are a multiple of R/S machine capacity.
- The sequence of storage requests can be changed according to the grouping way.

Notations:

k: the capacity of R/S machine.

p: the number of groups in which each group consists of exact k storage and k retrieval jobs.

S: the set of storage requests, with $|S| = n$ being the number of storage requests in S. $S = \{S_1, S_2, \ldots, S_n\}$, where S_i is the i-th storage request.

R: the set of retrieval requests, with $|R| = n$ being the number of retrieval requests in R. $R = \{R_1, R_2, \ldots, R_n\}$, where R_j is the j-th retrieval request.

O: the set of all requests, $O = S \cup R$.

B: buffer area where stored loads and retrieved loads are located (I/O location).

S_a^q, R_a^q: the set of storage requests and the set of retrieval requests in the a-th subgroup corresponding to the grouping way q (a solution of the grouping problem), respectively.

$G = \{G^1, \ldots, G^q, \ldots, G^m\}$: the set of solutions of the grouping problem, where G^q is the q-th solution of the grouping problem.

SG_a^q: the a-th subgroup in the q-th solution of grouping problem, ($SG_a^q \in G^q, a = 1, 2, \ldots, p$). $SG_a^q = \{S_a^q, R_a^q\}$.

SR_a^q: the set of possible routes in the subgroup SG_a^q.

$SR_{a_r}^q$: the subroute r in the set $SR_a^q, r = 1, 2, \ldots, Nr$, i.e. the sequence of storage and retrieval locations which the R/S machine has to visit to finish all requests.

t_{ij}: travel time from location i to location j, $i, j \in O$.

t_{bi}: travel time from the buffer location to the arbitrary storage location S_i, $S_i \in S$.

t_{jb}: travel time from the arbitrary retrieval location $R_j$to the buffer position, $R_j \in R$.

Ω_s^q: a solution of the sequencing problem.

$C(SR_{a_r}^q)$: cost of the subroute $SR_{a_r}^q$.

$TC(\Omega_s^q)$: total cost of the solution Ω_s^q.

For the problem was described in Sect. II-A, a solution of the problem is a set of the feasible subroutes $\Omega_s^q =$

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$\{SR^q_{1_r}, SR^q_{2_r}, \ldots, SR^q_{p_r}\}$ corresponding to the grouping way G^q, where $SR^q_{a_r} \in SR^q_a$, $a = 1, 2, \ldots, p$.

With the above notations, the cost of a subroute $SR^q_{a_r}$ is calculated by Eq. 3:

$$C(SR^q_{a_r}) = \frac{t_{bi}}{S_i \in S^q_a} + \sum_{S_i, R_j \in SG^q_a} t_{ij} + \frac{t_{jb}}{R_j \in R^q_a}, \quad (3)$$

Finally, the total cost of a solution Ω^q_s is computed by Eq. 4

$$TC(\Omega^q_s) = \sum_{a=1}^{p} C(SR^q_{a_r}). \quad (4)$$

The objective of solving the sequencing problem is to find a set of optimal subroutes $\Omega^{q^*} = \{SR^{q^*}_{1_{r^*}}, SR^{q^*}_{2_{r^*}}, \ldots, SR^{q^*}_{p_{r^*}}\}$ corresponding to the optimal grouping way q^* (or optimal solution of grouping problem) that minimizes the total traveling cost, is given by Eq. 5:

$$\min_{G_q \in G} TC(\Omega^q_s) \quad (5)$$

or

$$\min_{G_q \in G} \sum_{a=1}^{p} \min_r C(SR^q_{a_r}) \quad (6)$$

2) Routing of Storage and Retrieval Requests Problem: As indicated in [5], general optimal sequencing problem of a given requests list is NP-hard. This kind of problem shares the similar characteristics with delivery and pickup problem (DPP). Due to the high complexity of this problem, this work will not consider heuristics method for solving this problem with large sizes, instead, an exact method will be used to find the true optimal sequence of storage and retrieval requests in each group. The cases with higher capacity need to be further investigated in future researches.

Given a list of requests consists of k storage and k retrieval requests, the exact method to find the optimal sequence is expressed in the following steps:

- Step 1: List all possible sequences of all requests.
- Step 2: Compute cost function (travel time) of each possible sequence.
- Step 3: The sequence with minimal cost is selected as optimal sequence.

III. Local Search for Complete Grouping Problem

A. Solution Representation

Before constructing a feasible neighborhood of a solution (a step/walk), the representation of a solution need to be defined. For original grouping problem, several representations were introduced in [9], [10], [8]. For example, in [9], the authors presented *number encoding* and the *group encoding* which are used in genetic algorithm (GA). In this paper, a modified version of number encoding is presented associated with complete grouping problem (CGP) described in Sect. II-A.

A candidate solution to CGP is $2xn$ matrix, where n is the number of storage/retrieval requests in a block. The first row and second row are represented for the storage requests and the retrieval requests, respectively. Each row consists of a series of integer numbers in which each integer occurs k times (the capacity of R/S machine). In each row, the encoding scheme is represented as number encoding [8], the i-th element indicates the ID of the group to which item i belongs. Fig. 1 illustrates an example of this representation with R/S machine capacity $k = 2$ and the number of storage/retrieval requests are 10. It results in 5 groups, e.g. group 1 consists of fourth and seventh storage requests, and fifth and tenth retrieval requests, group 2 consists of third and sixth storage requests, and first and fourth retrieval requests, and so on.

Store	3	5	2	1	4	2	1	5	4	3
Retrieve	2	4	5	2	1	3	4	3	5	1

Fig. 1. An example of a solution representation.

B. Proposed Swap Strategies

Four swap strategies are presented to form a neighborhood of a solution, i.e. a walk or a move in local search.

1) Swap Strategy 1: In the first swap strategy, we will swap two storage genes and two retrieval genes randomly and independently. More specifically, two random genes are swapped in the storage row first and then in the retrieval row. The swap strategy is illustrated in Fig. 2 and performed as following steps:

- Swap two genes in store row:
  1) Randomly select two groups in store row (e.g 2,5).
  2) Randomly select a gene in each chosen group.
  3) Swap the two selected genes.
- Swap two genes in retrieval row by repeating 1, 2, 3.

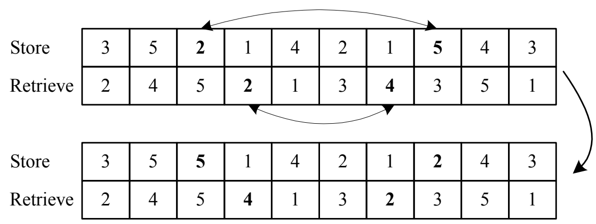

Fig. 2. Illustration of strategy 1.

2) Swap Strategy 2: In this strategy, we swap storage genes and retrieval genes together and an illustration is shown in Fig. 3 and performed as following steps:

- Randomly select a group from the store row and a group from the retrieval row (e.g. 1, 3).
- Randomly select a gene in each chosen group (e.g. underline number).
- Swap their columns, but keep in the same rows.

3) Swap Strategy 3:

- Randomly select two groups (contain both store retrieval items), (e.g. 1, 3).
- Randomly select one gene for each chosen group from the store row and one gene from the retrieval row (e.g. underline number).

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

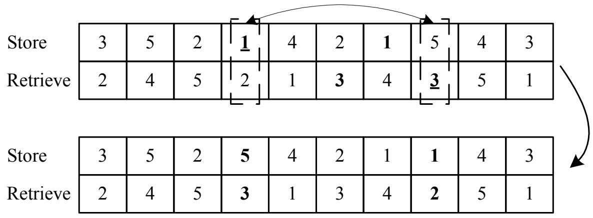

Fig. 3. Illustration of strategy 2.

- Swap the pair of genes between the two groups simultaneously.

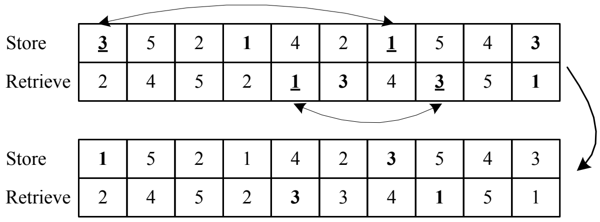

Fig. 4. Illustration of strategy 3.

4) Swap Strategy 4: One other way of doing the swap is to destroy a group and construct another group: All items in the randomly selected group are exchanged with items from other groups. This causes the greatest perturbation to the chromosome since it affects many other groups.

- Randomly select a groups for perturbation (e.g. underline-group 2).
- For each item in selected group, randomly select an item of the same type (store or retrieval) from another randomly selected group (e.g. bold numbers).
- Swap items in each pair.

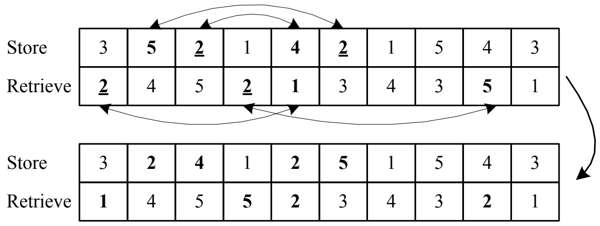

Fig. 5. Illustration of strategy 4.

C. Local Search Algorithms for the Problem

Iterated local search (ILS) is a metaheuristic method that try to improve a solution in an intelligent way by iterating local search in a number of times and using perturbation procedure. The procedure of ILS is shown in Alg 1 [11].

In this paper, a feasible initial solution was generated by randomly sample an integer series from 1 to p (p is the number of groups). The process is repeated k times (k is the capacity of R/S machine) to complete the generation of one row (storage or retrieval row). The second row is performed in similar way to form an initial solution. In this paper, we use the stochastic hill climbing (SHC) and random-restart hill climbing (RRHC) as local search (LS) engines. For the perturbation operator, the strategy modification of swap strategy 1 presented in Sect. III-B is used by swapping two genes twice instead of once in each row of the chromosome.

Algorithm 1 Iterated Local Search

loop
  s_0 = GenerateInitialSolution
  s^* = LocalSearch(s_0)
  repeat
    s' = Perturbation($s^*, history$)
    $s^{*'}$ = LocalSearch(s')
    s^* = AcceptanceCriterion$\Big(s^*, s^{*'}, history\Big)$
  until termination condition met
end loop

Finally, the acceptance criterion which determines whether a new solution is accepted or not as the new current solution need to be defined. The acceptance criterion controls the balance between intensification and diversification of searching process. Various acceptance criteria were introduced in [11] and their affects to the performance of ILS. In this work, the acceptance criterion *Better* [11] is used, it accepts a new solution as current solution if the new objective value is better than the current one, otherwise it keeps the current solution as new current solution.

IV. Experiments and Discussion

As mentioned in Sect. I the goal is aim to find the suitable operator, i.e. swap strategy and the local search engine for the formulated problem, various experiments were carried out to tackle these questions. In this work, due to the complexity of routing problem we just implemented for the capacity of R/S machine with dual-shuttle (capacity = 2). For problem instances, we use dataset from one aisle of an AS/RS with two rack arrays: the rack length of 80 meters and the rack high of 15 meters, and the total storage capacity is 2400. Different instances with the number of storage and retrieval requests of 10, 20, 30, 40 were generated by sampling randomly from all storage locations in an aisle. It should be noted that the dedicated storage strategy was used in these experiments. For algorithms, we use stochastic hill climbing, random restart hill climbing and iterated local search for searching the optimal/near optimal solutions of the problem. All of the local searches use the budget of 100000 function evaluations.

In the first experiments, we implemented the stochastic hill climbing equipped with four different swap strategies presented in Sect. III-B. On each problem instance, each local search was launched 15 times with different initial generated solution. In order to evaluate the performance of the proposed swap strategies, the median of 15 convergent profiles are used to compare four operators. The comparison results are shown in Fig. 6. In this figure, fifteen profiles of each local search are plotted in the same transparent color in the background and the median of the fifteen profiles is then plotted by a thick line. It is easy to observe that strategy 2 and 3 are worst than strategy 1 and 4, especially for small instance (instance 10, 20). For larger instance (30, 40) although strategy 3 started to converge faster than strategy 4 in between 500

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

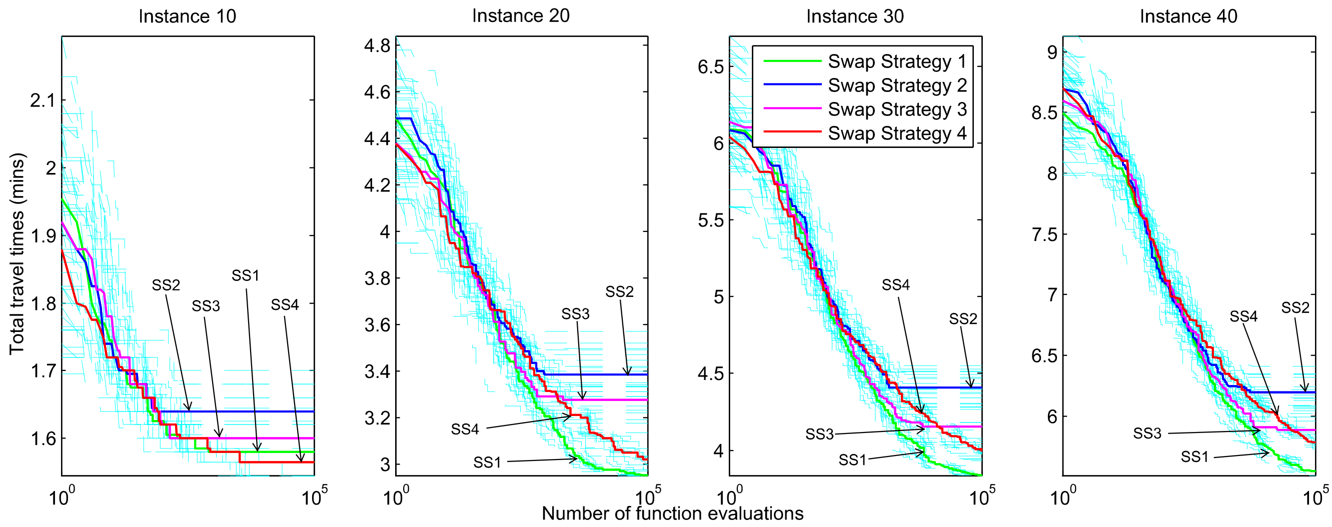

Fig. 6. Performance comparison of four proposed swap strategies with four instances of 10, 20, 30, 40 (SS: Swap Strategy).

and 8000 function evaluations, it was trapped in the local optimum after 10000 function evaluations. The figure shows an important observation that strategy 1 outperforms other strategies except in the case of instance 10 due to its small search space.

The second experiments were carried out with three local search algorithms: SHC, RRHC, and ILS in order to find the suitable algorithm for the formulated sequencing problem. It is shown that the strategy 2 is clearly worse than others while the strategy 3 still needs to be examined further for its performance. Therefore, we implemented the RRHC and ILS equipped with swap strategy 1, 3, 4 on the same four above instances of 10, 20, 30, 40. On each instance, the RRHC/ILS was also launched 15 times and recorded their profiles. Thereafter, the performances of these results were compared with the results of SHC obtained in the first experiments.

For restart mechanism, both RRHC and ILS use the fixed threshold of 1000 function evaluations. More specifically, if a solution cannot be improved after 1000 consecutive times, the RRHC restarts to a new random solution while ILS will perturb the current solution to a new solution that is not so far from the current local optimal solution. For the perturbation operator works with ILS, the modification of strategy 1 was used by swap two times in each rows of the chromosome.

Table I summarizes the medians of the best objectives (total travel times) over 15 runs for all algorithms and four instances. On the other hand, Fig. 7 and Fig. 8 give some insight into the performance of three LS algorithms with strategy 1 and strategy 3, respectively. Some main remarks can be drawn from these results. First, the convergent profiles in Fig. 8 along with the results in Table I confirm that the strategy 3 is worse than the strategy 1 and 4. Second, from Table I once again we could confirm that the swap strategy 1 outperforms the swap strategy 4. Third, for small instance of 10, RRHC and ILS perform better than SHC, however for larger instances (20, 30, 40) it is reserve, however these differences are not too much. These differences can be explained by two factors: (i) the perturbation operator is not strong enough to "kick" the solution out of local optimum or this local optimum is very near the true optimum which is very hard to obtain with the computational budget;

TABLE I
COMPARISON OF NINE LS ALGORITHMS: SHC, RRHC, AND ILS EQUIPPED WITH THREE SWAP STRATEGIES 1, 3, 4 ON FOUR INSTANCES.

Instance	Strategy	Total travel time (s)		
		SHC	RRHC	ILS
10	1	1.58	1.545	1.545
	3	1.6	1.545	1.545
	4	1.565	1.545	1.545
20	1	2.95	2.975	2.97
	3	3.275	3.005	2.975
	4	3.02	3.095	3.08
30	1	3.835	3.895	3.895
	3	4.155	3.925	3.87
	4	4	4.16	4.125
40	1	5.545	5.685	5.65
	3	5.885	5.775	5.645
	4	5.785	6.015	5.99

(ii) the threshold value for restart mechanism affects to the performances of RRHC and ILS.

V. CONCLUSIONS

The sequencing of storage and retrieval requests plays an important role in increasing throughput of AS/RS system, especially with multi-shuttle AS/RS system. However there is lack of a general formulation for this SP. In this paper, we have formulated and solved the sequencing of storage and retrieval requests in multi-shutte AS/RS system. The sequencing problem was decomposed into two subproblems: the first subproblem aims to find an optimal combination of storage and retrieval requests (grouping problem), and routing problem is the second subproblem which is find an optimal route for R/S machine to perform the requests in each group. We applied several LS algorithms for solving grouping problem while the exact method was used to solve routing problem. Through the computational experiment results, we concluded that swap strategy 1 is the suitable operator for LS algorithms since it outperforms others. About the LS algorithms, for small instances, RRHC and ILS proved its efficiency over SHC. However, with

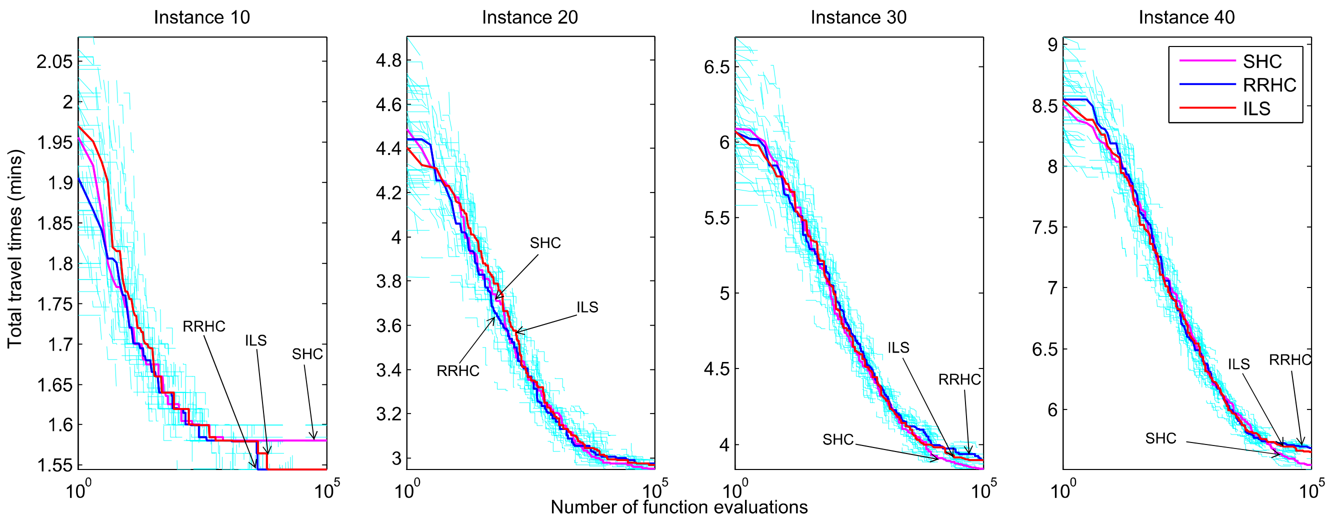

Fig. 7. Performance comparison of the SHC, RRHC and ILS equipped with swap strategy 1 on four instances of 10, 20, 30, 40.

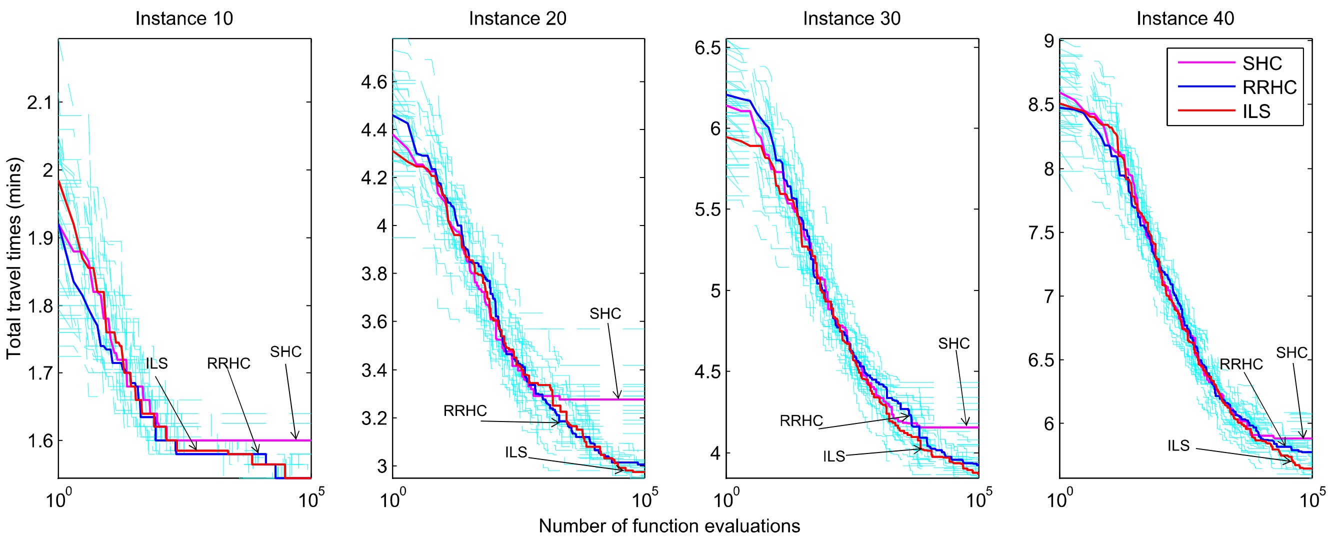

Fig. 8. Performance comparison of the SHC, RRHC and ILS equipped with swap strategy 3 on four instances of 10, 20, 30, 40.

larger instances SHC shows better performance than RRHC, ILS while ILS performs better than RRHC. This needs to be further investigated by proposing a more suitable perturbation operator to improve the performance of ILS.

References

[1] K. J. Roodbergen and I. F. Vis, "A survey of literature on automated storage and retrieval systems," *European Journal of Operational Research*, vol. 194, no. 2, pp. 343–362, Apr. 2009.

[2] H. F. Lee and S. K. Schaefer, "Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings," *International Journal of Production Research*, vol. 34, no. 10, pp. 2943–2962, 1996.

[3] H. F. Lee and S. K. Schaefer, "Sequencing methods for automated storage and retrieval systems with dedicated storage," *Computers & Industrial Engineering*, vol. 32, no. 2, pp. 351–362, Apr. 1997.

[4] J. P. van den Berg and A. (Noud) Gademann, "Optimal routing in an automated storage/retrieval system with dedicated storage," *IIE Transactions*, vol. 31, no. 5, pp. 407–415, 1999.

[5] M.-H. Han, L. F. McGinnis, J. S. Shieh, and J. A. White, "On sequencing retrievals in an automated Storage/Retrieval system," *IIE Transactions*, vol. 19, no. 1, pp. 56–66, 1987.

[6] Y. A. Bozer and J. A. White, "Travel-time models for automated Storage/Retrieval systems," *IIE Transactions*, vol. 16, no. 4, pp. 329–338, 1984.

[7] B. R. Sarker and P. Babu, "Travel time models in automated storage/retrieval systems: A critical review," *International Journal of Production Economics*, vol. 40, no. 23, pp. 173–184, Aug. 1995.

[8] A. Husseinzadeh Kashan, M. Husseinzadeh Kashan, and S. Karimiyan, "A particle swarm optimizer for grouping problems," *Information Sciences*, vol. 252, pp. 81–95, Dec. 2013.

[9] E. Falkenauer, "A new representation and operators for genetic algorithms applied to grouping problems," *Evol. Comput.*, vol. 2, no. 2, p. 123144, Jun. 1994.

[10] ——, "A hybrid grouping genetic algorithm for bin packing," *Journal of Heuristics*, vol. 2, no. 1, pp. 5–30, Jun. 1996.

[11] H. R. Loureno, O. C. Martin, and T. Stutzle, "Iterated local search," in *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science, F. Glover and G. A. Kochenberger, Eds. Springer US, Jan. 2003, no. 57, pp. 320–353.

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)