check for updates

MULTI-SHUTTLE AUTOMATED STORAGE AND RETRIEVAL SYSTEM

Amirhossein Mostofi[1+]
Hamidreza Erfanian[2]

[1]*School of Management, Victoria Business School, Victoria University of Wellington, New Zealand*
Email: mostofiamirhossein@gmail.com

[2]*Department of Mathematic, University of Science and Culture, Tehran, Iran*
Email: erfanian@usc.ac.ir

(+ Corresponding author)

ABSTRACT

Automated Storage and Retrieval Systems (AS/RS) are being widely used in the logistics industry. This typical System, Systems are computer controlled storage systems that can automatically store and retrieve loads. Major advantages of AS/RS include high throughput, efficient use of space, high reliability and improvement of safety. There are some different operational modes in typical system. The choice of operational mode depends on the storage location assignment policy. Shared storage approach is one of these policies. In this approach, the empty space created by an act of accumulation throughout recovery can be quickly filled which conduces to save time and money. This paper examines the joint optimization of (AS/RSs) scheduling in multi-shuttle AS/RSs under shared storage in fuzzy and dynamic environment. From the view of analytical model, the advantage of operational mode under shared storage is verified. For optimization multi-shuttle automated storage and retrieval system to minimize travel time, present a zero-one fuzzy mathematical programming model. Use Genetic algorithm (GA) with Matlab software to solve large-sized problems. Various numerical experiments are conducted to evaluate the performance of the proposed algorithm and investigate the impact of different parameters on computational efficiency. The result indicates that in matters where the primary objective is to reduce travel time cycle, shared storage policy is appropriate. Furthermore, the fuzzy sets theory can be an effective tool to model the uncertainties which are available around the real applications.

1. INTRODUCTION

Due to the merits of higher operational efficiency, multi-shuttle automated storage/retrieval systems (AS/RSs) become increasingly popular for storing products in factories and distribution centers. A typical multi-shuttle AS/RS has a series of storage aisles each of which is served by a multi-shuttle storage and retrieval (S/R) machine. The multi-shuttle S/R machine can carry multiple stock keeping units (SKUs) at a time, which implies that performing multiple operations (storage or retrieval) on a trip is feasible. The increase of storage and retrieval operations performed on a trip naturally leads to reducing the empty travel time of the S/R machine. Then the higher operational efficiency can be obtained compared with the single-shuttle AS/RS [1]. Nowadays companies need powerful and up-to-date tools to be able to survive competitions. Stock keeping systems are considered key components in the processes of production and storage of a product in manufacturing organizations and warehouses of large companies. Automated storage systems can play a decisive role in this regard. Manufacturers use fully automated stock keeping systems to improve accuracy and speed in performing storage and retrieval operations of products in the warehouses. Along with the technological progress in the present area, storage systems have grown

dramatically in recent years. The development of AS/RS is one of the most important advances made in modernization of industrial mechanisms. In the past, single-shuttle storage and retrieval systems were designed that could carry one pallet at a time. To promote systematic empowerment and capacity enhancement, multi-shuttle storage and retrieval systems were gradually introduced to the market. Planning methods applied to the use of this type of automated storage can vary. Due to that fact, various authors have suggested different methods for modeling and simulating the plan associated with such storage systems. So far, most of the literature is focused on discussing and evaluating multi-shuttle storage and retrieval systems. Also, several authors have outlined the uncertainty in modeling such systems and proposed fuzzy models. We will review some of these studies in the next part.

Storage location assignment is one of the important factors in a storage/retrieval scheduling problem, especially in multi-shuttle storage and retrieval systems. In 2011, Yang et al. examined location assignment problem (storage location and retrieval location) and used GA to solve it Yang, et al. [2]. Two years later, Yang et al raised the problem of the storage/retrieval scheduling and integral optimization of location assignment in multi-shuttle storage and retrieval systems. To solve their model, they proposed a two-phase Tabu Search algorithm for performing singular and multi-stage operation cycles [3]. Hwang et al. studied the order-picking problem in a single-aisle storage and retrieval system. They suggested an innovative algorithm that assigns classification to orders on each trip of S/R machine [4]. Elsayed and Lee described the classification procedure in a single-aisle system, aiming to minimize time lag [5]. Brezovnik et al. used the Ant Colony Algorithm to optimize multiple targets in a storage and retrieval system, taking several types of goods into account. They indicated that the type of storage and accumulation of loads depends on a variety of factors, of which they considered four: Factor of Inquiry (FOI); Product Height (PH); Storage and Space Usage (SSU); and Path to Dispatch (PD) [6].

Gambari et al. presented an analytical model for travel time calculation in automated storage and retrieval systems in which storage was based on ABC classification. They developed a numerical simulation model that worked in accordance with different scenarios [7]. Lerher et al. presented an optimization model for automated storage in which decision variables included minimizing travel time, minimizing costs and maximizing quality; annealing was used in their model too [8]. In 2011, Azizi et al. applied the Monte Carlo simulation to estimate travel time in a muli-shuttle system based on FEM standard and subsequently, compared different scenarios with one another [9]. Yang et al. used storage assignment optimization and sequencing automated storage/retrieval systems in shared storage locations. Thus, the resulting free space after the retrieval operation could be utilized more efficiently. To deal with large-sized instances, they applied an optimization algorithm to the largest travel distance [3].

This paper is organized as follows. Section 2 explains the characteristics of the problem; mathematical model and the introduction of GA are presented in Section 3 and Section 4, respectively. Computational results are included in Section 5. Finally, conclusions and hints for future work are presented in Section 6.

2. PROBLEM FORMULATION

We seek to propose an optimization model for automated storage and retrieval systems by the goals of its planning to minimize travel time in multi-shuttle storage/retrieval systems, using meta-heuristic algorithms. In line with the studies mentioned in the review of the literature, we also use fuzzy logic in this article to model uncertainties. Linear mathematical programming can be defined as the efficient allocation of limited resources to certain activities, to achieve pre-selected objectives.

Regarding uncertainty, fuzzy mathematical programming problems can generally be divided into the following categories:

- Mathematical programming that involves ambiguity in constraints.
- Mathematical programming that involves ambiguity in parameters.

- Mathematical programming that involves ambiguity in constraints and parameters.
- Mathematical programming that involves fuzzy variables.

The fuzzy mathematical model used in this paper belongs to the third category, commonly referred to as Robust fuzzy mathematical programming [10]. To clarify the matter, consider the fuzzy mathematical model below:

$$\min z = \sum_{j=1}^{n} \tilde{c}_j \, x_j \tag{1}$$

$s.t:$

$$\sum_{+ \, j=1}^{n} \tilde{a}_{ij} x_j \overset{\tilde{}^T}{\le} \tilde{b}_i \qquad i = 1, 2, \ldots\ldots, m, \tag{2}$$

$$x_j \ge 0 \qquad j = 1, 2, \ldots\ldots\ldots, n. \tag{3}$$

$\tilde{a}_{ij} \cdot \tilde{b}_j$ and $\tilde{c}_j$ are fuzzy numbers and $\widetilde{\min}$ and $\overset{\tilde{}^T}{\le}$ indicate fuzzy expansion (of min) and $\le$ fuzzy expansion, respectively. This model is called a Robust programming model. To solve a fuzzy mathematical programming problem, it is first converted into a normal mathematical programming problem, using mathematical theorems and propositions, and then it is resolved by the common problem-solving procedures in normal mathematical programming. In this paper, we ignore discussions regarding proof of such theorems. To learn more about the stages involved in converting a fuzzy mathematical programming problem into a normal mathematical programming problem. Given the explanations provided in the paper mentioned above, the corresponding normal mathematical programming problem of (1) is obtained as below:

$$\min z = \sum_{j=1}^{n} (c_j - \alpha \hat{c}_j) x_j \tag{4}$$

$s.t:$

$$\sum_{j=1}^{n} (a_{ij} + (\alpha - 1)\hat{a}_{ij})) x_j \le b_i + (1 - \alpha)\hat{b}_j \qquad i = 1, 2, \ldots\ldots, m, \tag{5}$$

$$x_j \ge 0 \qquad j = 1, 2, \ldots\ldots\ldots, n. \tag{6}$$

The value of the parameter α in the above model is a number belonging to the range $[0,1)$ which is chosen by the decision-maker. In addition that, this parameter is a kind of confidence level which means that the decision-maker with confidence level α wants to optimize the amount of the target function. In this paper, for the ease of modeling, and based on the article by Yang et al. several assumptions were necessary [1, 3].

The hypotheses of model are as follows:

- S/R machine can move in horizontal and vertical directions at the same time and under constant speed. We used Chebyshev distance time intervals between two stations. Here, Chebyshev distance points to maximum horizontal and vertical travel time.
- For different loads, the time needed for picking up and putting in is fixed and thus ignored in the model.
- The position and location of retrieval operations are already determined and therefore, identifiable.
- It is already clear whether the location of the initial activity is free or occupied by a certain load.
- In the present model, different types of loads have been taken into account and items can be placed in different locations. So, various items do not need different locations.
- Warehouse layout is not considered in this paper.
- The S/R machine is multi-shuttle.
- The proposed model is a fuzzy linear programming model.

3. MODEL FORMULATION

The following notations are used for mathematical formulation of proposed model:

Variables:

L Number of the shelves in the warehouse (shelves are numbered respectively as $1, 2, \ldots\ldots, L$)

LR Total number of requested retrieval locations

LE Total number of initial free locations

LA Total number of locations that may be visited during the cycle, for which we have $LA = LE \bigcup LR$

d_{ij} Travel time between i and j , so that $i, j \in \{0, LA\}$ and 0 represent the locations of arrival and departure.

x_{ij} Binary variable which is equal to 1 if it represents visits to point i as j th point in each cycle

The mathematical formulation of the proposed model is as follows:

$$\min \sum_{i \in LE} d_{0i} x_{i1} + \sum_{i \in AL} \sum_{j \in LR} \sum_{l=1}^{n} d_{ij} x_{il} x_{jl+1} + \sum_{i \in LR} d_{i0} x_{in+1} \tag{7}$$

$s.t:$

$$\sum_{i \in LE} x_{i1} = 1 \tag{8}$$

$$\sum_{i=2}^{n+1} x_{il} \leq 1 \qquad i \in LR, \tag{9}$$

$$\sum_{i \in LR} x_{il} = 1 \qquad l = 2, \ldots\ldots\ldots, n+1. \tag{10}$$

$$x_{il} \in \{0,1\} \tag{11}$$

The total time spent by S/R machine is calculated and minimized using the following rule. This rule consists of three parts: the first part relates to movements of the machine from entry/exit points to the first free location for storage. The second corresponds to movement of the machine along the aisle during the retrieval, and the third part deals with the movement of the S/R machine from the last retrieval location to the entry/exit points. Constraint (8) expresses that the first location to be visited should be one of the initial free locations. Constraint (9) holds that at the most, each location is visited only once per cycle. Constraint (10) determines that the requested retrieval locations will be visited in the order from the second to $(n+1)$ th point. Constraint (11) represents the range of model variables.

4. GENETIC ALGORITHM

GA is considered as an optimization and global search methods which is based on the simulated natural selection [11]. Holland developed GA in the 1970s. It is applied effectively to solve various combinatorial optimization problems and is based on probabilistic rules [12]. GA searches new and better solutions to a problem by improving the current population. The search is guided towards the principle of the survival of the fittest. This is obtained by extracting the most desirable characteristics from a generation and combining them to form the next generation. The population includes a set of chromosomes. Each chromosome in the population is a possible solution. The quality of each possible solution is measured by the fitness function. First, GA generates an initial population and then calculates the fitness value according to fitness function for each chromosome of the population. Fitness function is specifically generated for each problem. It may be simple or complex according to the problem. Then the optimization criterion is checked. If the optimization criterion is met, this solution can be considered as the best solution. Otherwise, a new population is regenerated using GA operators (selection, crossover, and mutation). According to their fitness values, chromosomes are selected for crossover operation using a selection operator. Therefore each chromosome will contribute to the next generation in proportion to its fitness. Then crossover and mutation operators are applied to the selected

population to create the next population. The process continues through generations until the convergence on optimal or near-optimal solutions. However, GA cannot guarantee to find the best optimal solution. GA operators are described as follows:

4.1. Population

It is a set of possible solutions to the problem. Since the size of the population varies according to the problem, there is no clear mark on how large it should be. Then, the fitness value for each chromosome of the population is calculated according to the fitness function.

4.2. Elitist Selection

Selection operator selects the chromosomes to be mated according to their fitness values. Elitist selection is used here which means that a practical variant of the general process of constructing a new population is to allow the best organism(s) from the current generation to carry over to the next, unaltered. This guarantees that the solution quality obtained by the GA will not decrease from one generation to the next.

4.3. Crossover

Crossover operator is a powerful tool for exchanging information between chromosomes and creating new solutions. It is expected that good parents may produce better solutions.

4.4. Mutation

This operator is used to prevent reproduction of similar type chromosomes in population. Mutation operator randomly selects two genes in chromosome and swaps the positions of these genes to produce a new chromosome. This technique is called swap mutation.

5. COMPUTATIONAL RESULTS

In this section, we evaluate the tractability of the proposed programming model regarding the solution quality and the required computation time. To do this, we perform some numerical experiments on a set of randomly generated problem instances in small and large sizes. The programming models are implemented in Lingo 11.0 modeling language. All experiments are implemented on a Laptop with a Core 5 Duo CPU processor and 4 GB of RAM.

5.1. Designing the Test Problems

Various test problems, with different sizes are considered to assess the performance of the proposed GA algorithm. We consider three sets of 9 small sized, 9 large sized problems to be solved using GA algorithm, i.e., a total of 18 instances were run. In each problem, the values of each group of parameters are generated randomly between their lower and upper bounds.

5.2. Setting the GA Parameters

Parameters of the proposed GA algorithm include *population size, crossover rate* and *mutation rate*. The values for *population size, crossover rate* and *mutation rate* are set to 200, 0.5, 0.1 200 and 100, respectively.

5.3. Computational Results

Summary of computational results for small sized problems is presented in Table 1. As mentioned in previous sections, a certain level of confidence for the decision maker is a real number that belongs to [0, 1) range which is chosen by the decision maker. So there are infinitely many choices to determine the amount of α. To better

understanding of the subject, examples are given for five different α values. Most of these examples can be solved by Lingo software or other exact methods in a reasonable time. By increasing the number of shuttles and Input or output (I/O) points, the calculation time is precisely unreasonable with accurate methods. Therefore, it seems that the number of shuttles has a significant impact on the calculation time. In the following, we will take a look at this.

Table 1 delineates the objective values for the miscellaneous amount of α for small sized problem. The 5th column of the table shows the objective values for $\alpha = 0.1$. This means that the decision maker with the given conditions is certain with the $\alpha = 0.1$ level of confidence that the minimum time will be obtained as the numbers listed in the table and related to the particular conditions. Also, the results of the table indicate that in most cases of the warehouse layout, the minimum duration decreases with an increase in the number of shuttles. This issue is quite reasonable regarding the reduction in the number of available cycles for R/S machine while increasing the number of shuttles.

Table-1. Objective values for different amount of α for small sized problem.

Problem number	a	b	n	Objective value				
				$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$
1	5	6	2	25.4	21.8	19.34	17.21	15.36
2	5	6	3	18.6	16.2	15.14	13.45	12.06
3	5	6	4	17	14	12.87	11.05	10.21
4	5	8	2	22.4	20.13	18.25	17.43	16.32
5	5	8	3	19.8	17.3	15.87	14.02	13.54
6	5	8	4	20	17.56	15.2	13.40	14.12
7	6	10	2	29.4	27.36	26.21	25.62	23.82
8	6	10	3	28.6	26.22	24.65	23.30	21.64
9	6	10	4	26.9	25.63	24.01	22.38	20.53

Source: Matlab software

Table-2. Objective values for different amount of α for large sized problem.

Problem number	a	b	n	Objective value				
				$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$
10	10	10	2	57.67	56.21	54.92	53.34	52.06
11	10	10	3	56.5	55.03	53.75	52.01	51.023
12	10	10	4	53.5	51.23	50.0	48.87	47.42
13	10	20	2	115.6	112.58	111.03	110.0	108.76
14	10	20	3	113.8	110.25	108.22	107.13	105.85
15	10	20	4	88.84	87.27	86.07	85.91	83.87
16	20	30	2	161.5	160.01	158.75	156.72	155.02
17	20	30	3	159.4	158.12	156.32	154.08	152.90
18	20	30	4	141.8	139.64	138.92	136.62	135.03

Source: Matlab software

The 6th column of the table is related to the objective values of $\alpha = 0.2$. Therefore, the decision maker will be certain with $\alpha = 0.2$ level of confidence that the minimum time will be obtained as the numbers listed in the table and related to the particular conditions. Also, the results of the table indicate that in most cases of the warehouse layout, the minimum duration decreases with increase in the number of shuttles. This issue is quite reasonable regarding the reduction in the number of available cycles for R/S machine while increasing the number of shuttles. The 7th, 8th and 9th columns of the table are concerned with the objective value of $\alpha = 0.3$, $\alpha = 0.4$ and $\alpha = 0.5$, respectively.

A summary of computational results for these examples is given in Table 2. Exact methods, especially Lingo optimization software, are not able to solve these examples at a reasonable time. In this section, five different values for α have been used for better understanding of the subject. The analysis of each table is similar to the analysis of small-sized instances. The 5th column of the table shows the objective values for $\alpha = 0.1$. This means that the decision maker with the given conditions is certain with the confidence level of $\alpha = 0.1$ that the minimum time will

be obtained as the numbers listed in the table and related to the particular conditions. Also, the results of the table indicate that in most cases of the warehouse layout, the minimum duration decreases with an increase in the number of shuttles. This issue is quite reasonable regarding the reduction in the number of available cycles for R/S machine while increasing the number of shuttles. The 6th column of the table is related to the objective values of $\alpha = 0.2$. Therefore, the decision maker will be certain with $\alpha = 0.2$ level of confidence that the minimum time will be obtained as the numbers listed in the table and related to the particular conditions. In addition, the results of the table indicate that in most cases of the warehouse layout, the minimum duration decreases with an increase in the number of shuttles. This issue is quite reasonable regarding the reduction in the number of available cycles for R/S machine while increasing the number of shuttles. The 7th, 8th and 9th columns of the table are concerned with the objective value of $\alpha = 0.3$, $\alpha = 0.4$ and $\alpha = 0.5$, respectively.

Table-3. Comparison between the performance of GA and VNS (run time) for small sized problem.

Problem number	a	b	n	Run time	
				GA	VNS
1	5	6	2	26.3	29.0
2	5	6	3	191.1	21.0
3	5	6	4	17.9	20.0
4	5	8	2	23.1	26.0
5	5	8	3	21.0	23.0
6	5	8	4	21.5	24.0
7	6	10	2	30.1	25.0
8	6	10	3	29.9	33.0
9	6	10	4	28.1	31.0

Source: Matlab software

Table-4. Comparison between the performance of GA and VNS (run time) for large sized problem.

Problem number	a	b	n	Run time	
				GA	VNS
10	10	10	2	59.2	64.0
11	10	10	3	57.7	97.0
12	10	10	4	54.1	53.0
13	10	20	2	117.4	101.0
14	10	20	3	115.6	166.0
15	10	20	4	90.1	147.0
16	20	30	2	163.1	212.0
17	20	30	3	161.6	187.0
18	20	30	4	143.1	117.0

Source: Matlab software

In previous sections, we stated that we proposed a model which is fuzzy with consideration of dynamics condition. Now, if we remove the dynamic condition and consider the $\alpha = 0$ in the defuzzied model, the model is identical to the proposed model in Yang, et al. [1]. In this case, we compared the solving methods. Yang et al. applied Variable Neighbourhood Search (VNS) approach to solve their proposed model. Table 3 and Table 4 compare the performance of our proposed GA with proposed VNS of Yang, et al. [1] (runtime) for small and large sized problem, respectively. Regarding the minimum target function, it is evident that the lower the cost of the optimal result, the success of the algorithm is greater in solving the model. The tables above show that the total calculated time using the proposed GA is less than VNS algorithm except for a limited number of exceptions.

6. CONCLUSIONS

So far, different methods and models have been proposed by various authors with the aim of optimizing the performance of automated storage and retrieval systems. Most of these models are designed with the assumption that the parameters are accurate. Presenting a model that exactly corresponds to reality is difficult and in some

circumstances, impossible. Therefore, the degree of closeness to reality can be considered as a criterion for evaluating the proposed models and procedures. In the present paper, the aim was to provide a model that is as close to reality as possible, given the extent to which it is compatible with our hypotheses. To this end, we made use of shared storage policy in a fuzzy environment. As a result of this paper, it can be said that concerning the reduced movements required to perform storage and retrieval operations, in cases where the primary objective is to reduce travel time cycle, the shared storage policy is an appropriate approach. The solved examples show that an increase in the number of shuttles leads to a reduction in travel time cycle. Although increasing the number of shuttles results in more costs in term of purchase and equipment of the warehouse, but given the consequent reduction in time and cost of performing operations, it will ultimately lead to profits for the warehouse. Also, it can be said that the Fuzzy set theory is an efficient tool to model ambiguities and uncertainties in such systems. As another result of this paper and based on the solved examples, it appears that when the storage is small, it is more appropriate to solve the proposed model with Lingo. On the other hand, the solved examples of large-sized problems, confirm the efficiency of the Genetic algorithm in solving the proposed model. By comparing the results of GA with the algorithms which are applied in the paper of Yang et al. 2015, it is observed that the minimum time obtained using the GA is less than those of the other two algorithms. Therefore, for the proposed model of the present paper, GA is more appropriate.

REFERENCES

[1]   P. Yang, L. Miao, Z. Xue, and B. Ye, "Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems," *Transportation Paper Part E: Logistics and Transportation Review*, vol. 79, pp. 164-177, 2015.Available at: https://doi.org/10.1016/j.tre.2015.04.009.

[2]   P. Yang, L.-X. Miao, and M.-Y. Qi, "Slotting optimization in a multi-shuttle automated storage and retrieval system," *Computer Integrated Manufacturing Systems*, vol. 17, pp. 1050-1055, 2011.

[3]   P. Yang, L. Miao, Z. Xue, and L. Qin, "An integrated optimization of location assignment and storage/retrieval scheduling in multi-shuttle automated storage/retrieval systems," *Journal of Intelligent Manufacturing*, vol. 26, pp. 1145-1159, 2015.Available at: https://doi.org/10.1007/s10845-013-0846-7.

[4]   H. Hwang, W. J. BAEK, and M.-K. Lee, "Clustering algorithms for order picking in an automated storage and retrieval system," *International Journal of Production Research*, vol. 26, pp. 189-201, 1988.Available at: https://doi.org/10.1080/00207548808947853.

[5]   E. Elsayed and M.-K. Lee, "Order processing in automated storage/retrieval systems with due dates," *IIE Transactions*, vol. 28, pp. 567-577, 1996.Available at: https://doi.org/10.1080/15458830.1996.11770701.

[6]   S. Brezovnik, J. Gotlih, J. Balič, K. Gotlih, and M. Brezočnik, "Optimization of an automated storage and retrieval systems by swarm intelligence," *Procedia Engineering*, vol. 100, pp. 1309-1318, 2015.Available at: https://doi.org/10.1016/j.proeng.2015.01.498.

[7]   M. Gamberi, R. Manzini, and A. Regattieri, *Analytical and numerical modeling of AS/RS cycle time in class-based storage warehousing. In Warehousing in the global supply chain*. London: Springer, 2012.

[8]   T. Lerher, M. Sraml, M. Borovinsek, and I. Potrc, "Multi-objective optimization of automated storage and retrieval systems," *Annals of the Faculty of Engineering Hunedoara*, vol. 11, pp. 187-194, 2013.

[9]   A. Azzi, D. Battini, M. Faccio, A. Persona, and F. Sgarbossa, "Innovative travel time model for dual-shuttle automated storage/retrieval systems," *Computers & Industrial Engineering*, vol. 61, pp. 600-607, 2011.Available at: https://doi.org/10.1016/j.cie.2011.04.015.

[10]   M. S. Pishvaee, J. Razmi, and S. A. Torabi, "Robust possibilistic programming for socially responsible supply chain network design: A new approach," *Fuzzy Sets and Systems*, vol. 206, pp. 1-20, 2012.Available at: https://doi.org/10.1016/j.fss.2012.04.010.

[11]   D. Goldberg, *Genetic algorithms in optimization, search and machine learning*. Reading: Addison-Wesley, 1989.

[12]   J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT Press, 1992.