Multiple Warehouses Scheduling Using Steady State Genetic Algorithms

Jalal Atoum[1] and Mamoun Al Rababaa[2]
[1]Computer Science Department, Princess Symaya University for Technology, Jordan
[2]Computer Science Department, Al Al-Bayt University, Jordan

Abstract: *Warehouses scheduling is the problem of sequencing requests of products to fulfill several customers' orders so as to minimize the average time and shipping costs. In this paper, a solution to the problem of multiple warehouses scheduling using the steady state genetic algorithm is presented. A mathematical model that organizes the relationships between customers and warehouses is also presented in this paper. Two scenarios of storage capacities (constants and varying capacities) and two strategies of search points (ideal point and random points) are compared. An analysis of the results indicates that multiple warehouses scheduling using the GENITOR approach with different warehouses capacities have better outcome than the usage of the traditional genetic algorithms).*

Keywords: *Multiple warehouse scheduling, genitor, ideal point search strategy, random point search strategy.*

Received October 27, 2008; accepted May 17, 2009

1. Introduction

The ability of firms to allocate customers to their available warehouses can be translated into a competitive advantage in current business environments. With its increasing significance, the problem of allocating a set of customers to multiple warehouses has begun to draw attention from both professionals and academicians. Scheduling is considered as one of the main issues in large systems. The appropriate scheduling will lead to the best exploitation of resources, lower cost and better customer demand satisfactions. Therefore, there is an urgent need to develop better automatic scheduling algorithms that enable us to achieve these goals. Evolutionary algorithms have achieved great success in the management process of scheduling in the following areas: Urban transit system, supply chain management processing factories, exam timetabling, flow shop manufacturing and job shop scheduling [2]. Specifically, Genetic Algorithms (GAs) as evolutionary algorithms had demonstrated impressive success in the area of warehouse scheduling with better results than other search algorithms [16].

2. Background

Multiple warehouse scheduling is the problem of sequencing requests for products (i.e., customer orders) so as to minimize the average time required to fill an order. This problem is complex due to several factors. First, the search space is quite large to determine all possible sequences of all orders. Several hundred possible orders are considered for each schedule. Optimization involves distinct performance measures, which can be inversely related. Search algorithms can be easily applied to multiple warehouse scheduling. However, given the expense of schedule evaluation and the number of possible schedules, knowledge poor search algorithms may be too costly. Heuristic based methods provide a promising alternative, as they explicitly leverage domain knowledge to directly construct candidate schedules. Approaches to scheduling have been varied and creative. Many points ranges from domain independent to knowledge intensive have been explored. Examples of the diverse underlying scheduling technique are local search [e.g., 11, 13], simulated annealing [e.g., 17], constraint satisfaction [e.g., 3], and transformational [e.g., 13]. In [11, 14] the authors have presented the mechanism of scheduling warehouses orders using standard GAs and a number of other search algorithms, they had proved the success of GAs in this area. The paper in [6] has proposed a GA for determining optimal replenishment cycles to minimize maximum warehouse space requirements. The results in this paper had showed that the proposed GA significantly outperforms a previously published heuristics. Zhou *et al.*[16] have addressed the scheduling of multiple stores using the standard GA with a proposed mathematical model. In [8, 9] the authors had integrated the evolutionary approach with domain-specific heuristics to obtain further improvements in computational requirements of the single, multi-objective and hybrid algorithms.

3. Genetic Algorithms

Genetic Algorithms (GAs) are based on the biological evolution processes that can be founded in natural evolution. In a GA generation after generation, the individual species compete with each other to survive (darwinian selection) [5]. Furthermore, GAs are evolutionary search and optimization algorithms that employ the mechanics of biological evolution for multi-objective problems [1, 3, 4, 12]. A GA presumes that a potential solution of any problem is an individual that can be represented by set of parameters. These parameters are regarded as the genes of chromosome and can be structured by a string of values in a binary form. The fitness value of an individual is used to reflect the degree of goodness of a solution (chromosome) for the problem [5]. When a constraint is violated a penalty is imposed on the individual timetable solution. The fitness of the individual solution depends in the penalties imposed by the constraint being violated [3, 10, 12]. The GA initially creates a population of solutions (individuals or chromosomes) and applies genetic operators to evolve the solutions from one generation to the next until it finds an optimal or near optimal solution, or terminates the execution under certain condition without finding any solution. The most important genetic operators that have been proposed by Holland to reproduce a new solution are [7]:

- Selection operator: Select chromosomes from the population according to their fitness values and called them parents.
- Crossover operator: Produce new offsprings by interchanging subparts of the selected chromosomes.
- Mutation operator: Randomly flips some bits in a new offspring.

Genetic algorithms have gained a primary importance in the field of scheduling, together with other stochastic methods as tabu search and simulated annealing. In general, GAs are used in scheduling for searching iteratively the best path in the tree of all possible decision sequences, which is often too wide for exact searches. However, GAs are not well suited for fine tuning structures, so that variations improving local search efficiency are frequently added.

4. Genitor Algorithm

Genitor is a GA and it is known as steady state genetic algorithm. It is one of the most effective genetic algorithms [15]. Genitor works in the following way:

a. Compute the fitness value for each chromosome (solution) in the generation (denoted by Sf).
b. Compute the average fitness value of all the chromosomes in the generation (denoted by Pf).
c. Each chromosome will be given a Rank greater than one if its fitness value is greater than the average fitness and a Rank less than one if its fitness value is less than the average fitness.
d. All chromosomes that have a rank greater than one will be subjected to the crossover process.
e. Crossover one pair of chromosomes who have the highest ratio:
f. $$FR = \frac{Sf}{Pf}$$
g. The offspring produced from the crossover will replace the chromosome that has the lowest fitness value in the current generation.
h. The above steps are repeated until the stopping condition is satisfied.

5. Multiple Warehouses Scheduling Through GAs

Multiple warehouses scheduling through traditional GAs have addressed the following points:

a. Clarifying the links between customers and warehouses, by arranging a group of customers in the contract symbolizes by *V* and set of warehouses that is symbolized by *U*. Finally, a graph that links customers with warehouses is formed. An example of such a graph is shown in Figure 1.

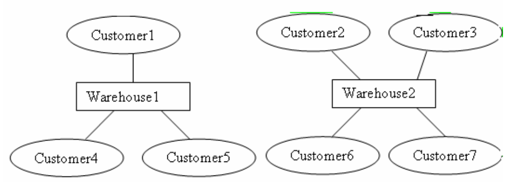


Figure 1. The distribution of customers orders on a number of warehouses.

b. Proposing a mathematical model to simulate the management of scheduling multiple warehouses. In addition, this model should specify the relationship between customers and warehouses. This model has focused on the following points:
   - Finding a mathematical function for calculating the fitness value effectively.
   - Proposing two strategies one to define the scope of the search space and the other to calculate weights of the goal fitness value.
   - Applying the GA using this proposed mathematical model to find the best solutions in the scheduling process.

In this paper the GA that is used for multiple warehouses scheduling is outlined below:

Begin
t=0

Initialize the population of parents P(0)
Determine the set of non _ dominated solutions E(0)
While (not termination condition) do
  Begin
    Recombine P(t) to yield the population of offspring C(t)
    Modify the population of P(t) and C(t)
    Update the set of non _dominated solutions E(t)
    Evaluate C(t) by Ideal Point Strategy or Random Points Strategy
    Select P(t+1) from P(t) and C(t)
    t=t+1
  End
End

where *t* is a generation number counter, *P(t)* is a generation (i.e., a set of solutions) in iteration *t, C(t)* is a off springs that are resulted from crossover in iteration *t*, and *E(t)* is a set of best solutions obtained in iteration *t*.

The quality of the solution generated for the multiple warehouse scheduling problem must be identified using two objectives; the time and the cost of shipping. Therefore, the GA proposed above to solve this problem should produce non-dominant solutions. A solution is called non-dominant solution, if there exists no other solution for which at least one of its objectives has a better value while values of remaining objectives are the same or better.

6. Scheduling Mathematical Model

In this paper, the subsequent mathematical model is used to represent the multiple warehouses scheduling problem [16]. Let us define the following symbols:

V: represents the *m* customers.
U: represents the *r* warehouses.
E: the Edges that link customers with warehouses. For each edge *Eij* (link warehouse *i* with customer *j*).
V_i ; Customer *i* request.
q_i: Warehouse *j* capacity.

The criteria for evaluation depend on two main factors: the shipping cost c_{ij} between warehouses *i* and the customer *j* and the shipping time spent $T_{ij:}$ in the process of shipping from warehouses *i* to the customer *j*. These two factors are represented in forms of two objective functions to solve the problem of scheduling multiple warehouses as follows:

1. An objective function to minimize the shipping cost c_{ij} of request V_i of every customer *i*, (i=1..m), from the warehouse *j*, (j=1...r), that is assigned to him (i.e., the only one link x_{ij} that is equal to 1):

$$\text{Minimize } f_1(x) = \sum_{i=1}^{m}\sum_{j=1}^{r} v_i c_{ij} x_{ij}$$

2. An objective function to minimize the shipping time $T_{ij:}$ of every customer *i*, (i=1...m), from the warehouse *j*, (j=1...r), that is assigned to him (i.e., the only one link x_{ij} that is equal to 1):

$$\text{Minimize } f2(x) = \sum_{i=1}^{m}\sum_{j=1}^{r} tijxij$$

subject to:

a. Every customer *i* has been assigned to only one warehouse j.

$$\sum_{j=1}^{r} xij = 1 \qquad , i=1,2,\ldots m$$

$$xij = \begin{cases} 1 & \text{if customer } i \text{ is allocated to} \\ & \text{warehouse } j, i = 1,2,..m, j = 1,2,..r \\ o & \text{otherwise} \end{cases}$$

b. The total demands of all customers V_i s, (i=1...m), does not exceed the total capacities of all warehouses.

$$\sum_{i=1}^{m} vixij \le qi \qquad , i=1,2,\ldots m$$

Merging the shipping cost and the shipping time objective functions yield the following fitness function:

$$f(x) = w1.f1(x) + w2.f2(x)$$

where w_1 is a fixed weight of f1(x), w_2 is a fixed weight of f2(x).
The way to determine the values of w_1 and w_2 will be explained later in this section.
The evaluation of the fitness function f(x) is computed as follows:

$$eval(f(x))=w1.f'1(x)+w2.f2'(x)$$

where $f1(x)$ is the first derivative of $f_1(x)$, and $f2'(x)$ is the first derivative of $f_2(x)$.
The first derivative $f'_1(x)$ (shipping cost function) is computed as follows:

$$f'1(x) = \frac{f1(x) - f1,\min(x)}{f1,\max(x) - f1,\min(x)}$$

where $f_{1,max}(x)$ is the highest cost value achieved for any chromosome in a generation, and $f_{1,min}(x)$ is the lowest cost value achieved for any chromosome in a generation. The first derivative $f'_2(x)$ (shipping time function) is computed as follows:

$$f'2(x) = \frac{f2(x) - f2,\min(x)}{f2,\max(x) - f2,\min(x)}$$

where $f_{2,\max}(x)$ is the highest time value achieved for any chromosome in a generation, and $f_{2,\min}(x)$ is the lowest time value achieved for any chromosome in a generation. Finally, the values of w_1 and w_2 are computed as follows:

$$w1 = \frac{\underline{w1}}{\underline{w1} + \underline{w2}} \qquad w2 = \frac{\underline{w2}}{\underline{w1} + \underline{w2}}$$

where $\underline{w1}$ *is* $f'1(x) - f'1, min(x)$, and $\underline{w2}$ *is* $f'2(x) - f'2, min(x)$.

7. Scope of Search Space

In this paper, the scope of the search space is defined in two strategies:

1. Ideal Point strategy.
2. Random Points strategy.

A detailed explanation of each strategy is introduced bellow.

7.1. Ideal Point Strategy

The scope of the search space in this strategy is done by calculating the values of both $f'_1.min(x)$, $f'_2.min(x)$ which constitute the ideal point value (Ideal Point) in a given generation. This in turn leads to the observation that the best point for search is the point of idealism [11]. This strategy is defined as follows:

eval(f(x))=w1.f'1(x)+w2.f2'(x)

where $f'_1(x)$, $f'_2(x)$, w_1, and w_2 are as defined above.

7.2. Random Points Strategy

The scope of search in this strategy is done through a number of random variables. This will ultimately enable us to get more than one point of search that are distributed in all regions of possible solutions in a given generation. This in turn increases the effective implementation of the GA to find the best solution [11]. This strategy is defined by eval(f(x)) as described above. However, w_1 and w_2 are given random values in which their summation is equal to one. Table 1 compares these two strategies. Whereas, Figures 2 and 3 illustrate graphically these two strategies:

Table 1. Ideal point and random points strategies comparisons.

	Ideal Point Strategy	**Random Points Strategy**
Search Space	Limited scope of the search at a certain point	the scope of the search include many points in the surrounding area.
W_1 and w_2 calculation	Requires calculating weights w_1 and w_2 using several mathematical formulas	assign random values to w_1 and w_2

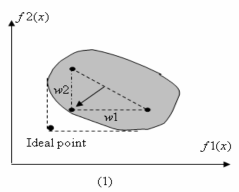


Figure 2. Ideal point strategy.

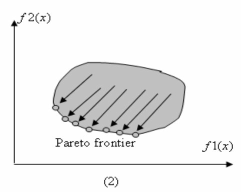


Figure 3. Random points strategy.

8. Penalty Function

Given that the capacity is specified for each of the warehouses, some situations may arise in which the demands of some customers can not be satisfied or partially satisfied by such capacities. These situations are addressed by penalizing the fitness value. Consequently, the fitness function defined earlier has to be amended by multiplying it by a penalty as follows:

eval(f(x))=p(x).(w1.f'1(x)+w2.f2'(x)), such that *P(x)* is a function of punishment and it is defined as follows:

$$P(x) = \begin{cases} \alpha + (\sum_{i=1}^{m} v_i x_{ij} - q_i)/q_i & \text{if } \sum_{i=1}^{m} v_i x_{ij} > q_i, \forall j \\ 1 & \text{if } \sum_{i=1}^{m} v_i x_{ij} - q_i) \le q_i, \forall j \end{cases}$$

where v_i is the quantity needed by customer i, and q_j is warehouse j capacity , and α is Variable value such that $1 \le \alpha \le 2$. In the case that all requests of a customer are satisfied, then P(x) is equal to 1.

9. Constraints

a. Each chromosome must represent all customers none redundantly.
b. Each customer must be assigned to a warehouse. A warehouse can be assigned to more than one customer.
c. In the case of deficiency in a customer satisfaction from his assigned warehouse, a search should be done to find another warehouse that still has a capacity to satisfy the rest of this customer request.
d. The fitness value is computed in two ways, once using the Ideal Point strategy and in the second using the Random Points strategy.
e. In the case of exhaustion of warehouses of their capacities without satisfying all customers' requests, then the penalty function is applied as explained earlier.

10. Analysis of the Results

In this paper, the following constants are used:

1. Population size = 200.
2. Number of generations = 500 .
3. Crossover rate = 0.4-0.2.
4. Mutation rate = 0.6-0.4.
5. Punishment values= 1.6-1.5.
6. Number of warehouses = 7.
7. Number of customer = 7.

Table 2. Shipping cost between customers and warehouses.

Customer i	Warehouse j							Total Demand (in units)
	1	2	3	4	5	6	7	
1	2.9	3.2	3.5	3.14	3.15	3.0	2.1	113.644
2	3.9	4.0	4.3	3.62	3.60	4.1	5.8	25.360
3	3.5	3.6	3.5	3.14	3.12	3.6	4.8	82.507
4	3.5	3.6	3.6	3.19	3.17	3.6	4.9	80.159
5	3.3	3.4	3.0	2.99	3.07	3.4	4.6	75.274
6	3.3	3.4	3.1	3.04	3.13	3.5	4.7	116.064
7	3.1	3.3	3.8	3.31	3.28	3.2	1.8	329.263

Table 3. Shipping time between customers and warehouses.

Customer i	Warehouse j						
	1	2	3	4	5	6	7
1	4.48	5.83	7.90	7.33	8.08	4.18	2.11
2	15.91	14.75	12.23	12.21	12.03	15.26	19.23
3	10.86	9.70	7.33	7.15	6.98	10.21	15.45
4	11.65	10.48	8.11	7.93	7.76	11.00	16.23
5	9.13	7.96	4.85	5.73	6.58	8.58	14.61
6	9.41	8.25	5.30	6.18	7.11	9.01	15.06
7	6.48	7.40	9.91	9.33	8.73	6.20	0

Two scenarios of warehouses capacities were used in the analysis; in the first scenario all warehouses have the same capacities, while in the second scenario each warehouse has a capacity of 10% greater than its neighbor warehouse. The system is executed twice using the Ideal Point strategy for each of these two scenarios. Another, execution of the system is also done twice using the Random Points strategy for each of these two scenarios. Table 2 presents the shipping costs between any customer and any warehouse, While Table 3 presents the shipping times between any customer and any warehouse.

11. Analysis of Results Using Traditional GA

The following two sections present the results of executing the system. The first section presents the results that are based on the conventional GA, while the second section presents the results that are based on the Gintor GA. Traditional GA Warehouse scheduling results:

Executing the multiple warehouses scheduling system using the traditional GA has produced the results that are shown in Figure 4. This figure provides a comparison between the results of the fitness values *eval(f(x))* achieved by the multiple warehouses scheduling for each of storage capacities scenario, and for the two search strategies mentioned earlier. In the reset of this paper, the following abbreviations will be used:

- Eq-Str1: Scheduling with equal warehouses capacities using Ideal Point Strategy.
- Eq-Str2: Scheduling with equal warehouses capacities using Random Points Strategy.

Dif-Str1: Scheduling with different warehouses capacities using Ideal Point Strategy.
Dif-Str2: Scheduling with different warehouses capacities using Random Points Strategy.
From the graph of Figure 4, the results are noticed as in Table 4.

Table 4. Results of Figure 4.

Scheduling Policy	The Chromosome that Satisfy the Highest Fitness	The Highest Value of the Fitness Function
Eq-Str1	81	1.603327
Eq-Str2	92	3.96114
Dif-Str1	88	1. 6432
Dif-Str2	95	4.09554

Therefore, we can observe that multiple warehouses scheduling using Random Points strategy has better results regardless of the capacity scenario used. Furthermore, multiple warehouses scheduling with various capacities has better results than with equal warehouse capacities.

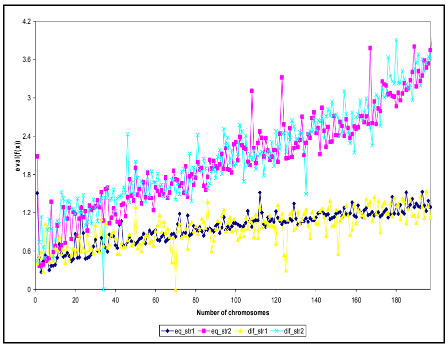


Figure 4. Results of the evaluation function using traditional GA.

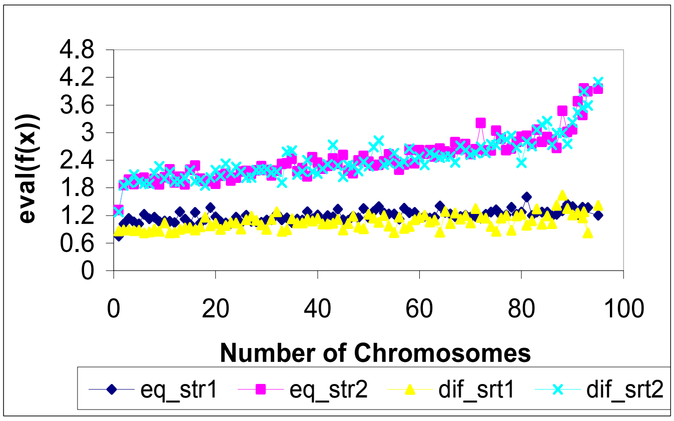


Figure 5. Results of the evaluation function using genitor GA.

Genitor Warehouse scheduling results: executing the multiple warehouses scheduling using the GENITOR has produced the results that are shown in Figure 5. The chart shown in this figure provides a comparison between the results of the fitness values eval(f (x)) achieved by multiple warehouses scheduling for each of storage capacities scenario using the

From the graph of Figure 5, the results are noticed as in Table 5.

Table 5. Results of Figure 5.

Scheduling Policy	Chromosome # that Satisfy the Highest Fitness Value Using Traditional GA	Highest Fitness Value Using Traditional GA	Chromosome # that Satisfy the Highest Fitness Value Using GENITOR	Highest Fitness Value Using GENITOR
Eq_str1	112	1.51508	81	1.603327
Eq_str2	189	3.80388	92	3.96114
Dif_str1	195	1.53629	88	1. 6432
Dif_str2	180	3.91	95	4.09554

12. Conclusions

Evolutionary Algorithms have played a major role in finding optimal solutions to scheduling problems that takes into considerations the exploitation of time and cost of shipping as key factors to prove their effectiveness. In this paper, the authors have provided a multiple warehouses scheduling system that uses a mathematical model through Genitor. The results of this system have indicated faster time and better cost of shipping than the result of a similar system that uses the traditional GA. According to the results obtained in this paper, it has been observed that the best solution to the scheduling problem has been achieved using the Genitor with varying warehouse capacities using the Random Points strategy. Also, it has been concluded that the results of using the Genitor algorithm has outperformed the results of using the traditional GA regardless of the warehouses capacities scenario and the search strategy used.

References

[1] Heong A. and Yun L., "Multi-Objective Benchmark Studies for Evolutionary Computation," *in Proceedings of 2001 Genetic and Evolutionary Computation Conference*, California, USA, pp. 393-396, 2001.

[2] Jose A. and Pablo M., "Approaches Based on Genetic Algorithms for Multiobjective Optimization Problems," *in Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, (eds), in Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 3-10, USA, 1999.

[3] Stefan B., Martin B., Thiele L., and Zitzler E., "Multiobjective Genetic Programming, Reduing Bloat Using SPEA2," *in Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, vol. 1, pp. 536-543, New Jersey, 2001.

[4] Lino C. and Pedro O., "An Evolution Strategy for Multiobjective Optimization," *in Proceedings of Congress on Evolutionary Computation (CEC'2002),* IEEE Service Center, vol. 1, pp. 97-102, New Jersey, 2002.

[5] Man, K., Tang K., and Kwong S., *Genetic Algorithms Concept and Design*, Spring-Verlag, 1999.

[6] Ming-Jong Y. and Weng-Ming C., "A Genetic Algorithm for Determining Optimal Replenishment Cycles to Minimize Maximum Warehouse Space Requirements," *Omega ISSN 0305-0483 CODENOMEGA6*, vol. 36, no. 4, pp. 619-631, 2008.

[7] Mitchell M., *An Introduction to Genetic Algorithms*, *A Bradford Book*, MIT Press, 1998.

[8] Naso D., Turchiano B., and Meloni C., "Single and Multi-Objective Evolutionary Algorithms for the Coordination of Serial Manufacturing Operations," *Journal of Intelligent Manufacturing*, vol. 17, no. 2, pp. 249-268, 2006.

[9] Naso D., Surico M., Turchiano B., and Kaymak U., "Genetic Algorithms for Supply Chain Scheduling: A Case Study on Ready Mixed

Concrete," *European Journal of Operation Research*, vol. 177, no. 3, pp. 2069-2099, 2007.

[10] Pinedo M., *Scheduling; Theory, Algorithms, and Systems*, Prentic Hall, 2002.

[11] Rana S., Howe A., Mathias K., and Whitley D., "Comparing Heuristic, Evolutionary and Local Search Approaches to Scheduling," *in Proceedings of the Third Artificial Intelligence Planning Systems Conference (AIPS'96*), AAAI Press, pp. 174-181, 1996.

[12] Ross P., Hart E., and Corne D., "Genetic Algorithms and Timetabling," *in A. Ghosh and K. Tsutsui (eds), Advances in Evolutionary Optimisation*, Springer, 2003.

[13] Smith D. and Parra E., "Transformational Approach to Transportation Scheduling," *in M. Burstein (Editor), ARPA/Rome Laboratory Knowledge Based Planning and Scheduling Initiative Workshop proceedings*, pp. 205-216, Palo Alto, CA, USA, 1994.

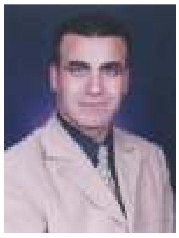
[14] Whitley L., Howe A., Rana S., Watson J., and Barbulescu L., "Comparing Heuristic Search Methods and Genetic Algorithms for Warehouse Scheduling, Systems, Man, and Cybernetics," *in Proceedings of International Conference on Systems*, vol. 3, no. 11-14, pp. 2466-2471, San Diego, 1998.

[15] Whitely L. and Kauth D., "Genitor: A Different Genetic Algorithm," *in Proceedings of the Rocky Mountain Conference on Artificial Intelligence*, vol. 2, pp. 118-130, 1988.

[16] Gengui Z., Min H., and Gen M., "A Genetic Algorithm Approach to the Bi-Criteria Allocation of Customers to Warehouses," *International Journal of Production Economics*, vol. 86, no. 1, pp. 35-45, 2003.

[17] Zweben M., Daun B., Davis E., and Deale M., "Scheduling and Rescheduling with Iterative Repair," *in Monte Zweben and Mark Fox (Editors), Intelligent Scheduling*, pp. 241-255, Morgan, 1994.

Jalal Atoum is an associate professor in the Computer Science Department at Princess Symaya University for Technology, Jordan. He received his BS degree in computer science from Yarmouk University, Jordan in 1984. He received his master degree in computer science from University of Texas at Arlington, USA in 1987. Received his PhD in computer science from University of Houston, USA in 1993. His interest research areas includes artificial intelligence, databases, object oriented programming languages, and distributed systems.



Mamoun Al Rababaa received MSc degree in 1995 and his PhD in 1999 in the area of computer engineering from Ukraine. His research interests include areas like information security, network security, internet security, viruses and artificial intelligence, expert systems, and genetic algorithms.