

УДК 519.78

Application of Self-Configuring Genetic Algorithm for Human Resource Management

Andrej Škraba,* Davorin Kofjač[†]
Anja Žnidaršič,[‡] Matjaž Maletič[§]

Faculty of Organizational Sciences
University of Maribor
Kidričeva cesta, 55a, SI–4000 Kranj
Slovenia

Črtomir Rozman[¶]
Faculty of Agriculture and Life Sciences
University of Maribor
Pivola, 10, SI–2311 Hoče
Slovenia

Eugene S. Semenkin,^{||} Maria E. Semenkina^{**}

Vladimir V. Stanovov^{††}
Siberian State Aerospace University
Krasnoyarsky rabochy, 31, Krasnoyarsk, 660014
Russia

Received 10.11.2014, received in revised form 02.12.2014, accepted 15.12.2014

This paper describes the problem of human resource management which can appear in many organizations during restructuration periods. The problem is simulated by a dynamic model, similar to a supply chain model with several ranks. The problem of finding the optimal combination of transition coefficients, including the fluctuation coefficients, is transformed into an optimization problem. To solve this problem, a self-configuring genetic algorithm is applied with several constraint handling methods. Additional constraints are defined in order to avoid undesirable oscillations in the system. The results show that this problem can be efficiently solved by the presented methods.

Keywords: human resources management, simulation, genetic algorithm, constrained optimization, self-configuration.

Introduction

The present paper describes the continuation of the research in the field of human resource management [1] where the problem of modelling and optimal control of the restructuring process

*andrej.skraba@fov.uni-mb.si

†davorin.kofjac@fov.uni-mb.si

‡anja.znidarsic@fov.uni-mb.si

§matjaz.maletic@fov.uni-mb.si

¶crtomir.rozman@um.si

||eugenesemenkin@yandex.ru

**semenkina88@mail.ru

††vladimirstanovov@yandex.ru

© Siberian Federal University. All rights reserved

in a hierarchical management system for eight topmost ranks was addressed. Since new standards for the structure of the system were issued, the goal was to achieve new desired values, i.e. number of officers in a particular rank in the shortest possible time. An additional condition is that oscillations in the promotions should be avoided. The problem was addressed in several previous studies [2, 3], nevertheless the dynamic character of the problem as well as time variant boundaries were not adequately addressed.

The problem of human resource management is similar to the problem of supply chain management therefore the results provided in the present paper could also be applied in the field of supply chains. Efficient control of the supply chain is one of the main concerns of cost-effective enterprise management. Since the basic structure of the supply chain consists of cascaded, exponential delay elements it will inevitably examine the oscillations known as the bullwhip or Forrester effect [4]. It is important to mention that the problem of oscillations in the control of human resource management as well as in the supply chain is an important topic of current research efforts [5, 6].

In this paper, a novel approach to minimize the oscillations in the control strategies based on evolutionary computations will be presented providing promising results with possible applications not only in the field of human resource management but in the field of supply chain management as well.

As an optimization technique in this paper we used a self-configuring genetic algorithm. The genetic algorithm is a stochastic global search method which simulates the model of natural evolution. This algorithm was chosen as it does not require any special information about the goal function and can effectively solve problems with algorithmically given objectives and constraints.

1. Modelling of discrete strictly hierarchical human resource transition system

In the strictly hierarchical manpower model, we suppose that promotion to a higher rank is only possible from the rank which is one step lower. There is no possibility, for example, to jump from rank x_1 to rank x_3 ; the only possible transition would be $x_1 \rightarrow x_2 \rightarrow x_3$. Fig. 1 shows the structure of the manpower system with eight elements as the case study. The input recruitment to the system is represented by $\mathbf{u}(k)$. The transitions between particular ranks x are determined by functions R , which depend on the number of men at the particular rank x and the transition coefficient r . The last transition $R(x_8, r_8)$ represents retirement. Wastage is determined by functions F , which are also dependent on the number of humans at a particular rank and the fluctuation coefficients f .

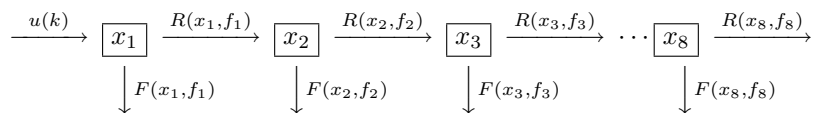


Fig. 1. Structure of manpower system with eight ranks, x_1 to x_8

The function $R(x_n, r_n)$ represents the transition function from a lower to a higher rank. The function value is determined by the number of men at the particular rank x_n , and transition coefficient r_n . Similarly, $F(x_n, f_n)$ represents the wastage (fluctuation) function from a particular

rank. The function value is determined by the number of humans at the particular rank x_n , and fluctuation coefficient f_n . The system is defined in a discrete state space as:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \end{cases} \quad (1)$$

where the matrix of coefficients \mathbf{A} is as follows:

$$\mathbf{A} = \begin{bmatrix} 1 - s_1(k) & 0 & 0 & \dots \\ r_1(k) & 1 - s_2(k) & 0 & \dots \\ 0 & r_2(k) & 1 - s_3(k) & \dots \\ 0 & 0 & r_3(k) & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (2)$$

where net growth factor $s_n(k) = r_n(k) - f_n(k)$; $n = 1, 2, \dots, 8$. The input $u(k)$ in the considered system is provided by coefficients of the matrix \mathbf{B} to x_1 where the dynamics of the inflow to the first rank is described as:

$$x_1(k+1) = [1 - s_1(k)]x_1(k) + u(k). \quad (3)$$

The system represents a cascaded exponential delay chain with outflows from particular ranks which in general exercises the exponential response when triggered by the input value, i.e. recruitment $u(k)$.

2. Case study example

Fig. 2 shows the values of our test example from the previous study [1], where we consider the current and desired numbers for each of eight considered ranks. The x -axis represents the number of humans and the y -axis represents eight ranks. The initial values are shown in the left column. The current value is shown on the graph by the solid-line rectangle while the desired values are represented as dashed rectangles. It can be observed that at some ranks, the desired values are below the current values and at others vice versa. Since the considered system represents a hierarchical chain, it represents an additional challenge at the system control. Therefore the prescribed task is to set the values of humans at a particular rank to the desired values, which might be below, above or equal to the current value.

The criteria function which determines the deviation of the number of men at particular ranks from the desired values is stated as:

$$J = \sum_{n=1}^r \sum_{i=0}^{t_q} (z_n(i) - x_n(i))^2, \quad (4)$$

where $z_n(i)$ is the target function value for the rank n at the step i . Eq. (4) represents the summation of the squared deviation from the target for the prescribed time horizon t_q as well as for the number of ranks r . Therefore we should determine the minimum deviation with respect to time and ranks as $\min_{u \in U, r \in R, f \in F} J$ where U , R and F are sets of input parameters subject to constraints:

$$\begin{aligned} \mathbf{u}_{min}(k) &\leq \mathbf{u}(k) \leq \mathbf{u}_{max}(k), \\ \mathbf{r}_{min}(k) &\leq \mathbf{r}(k) \leq \mathbf{r}_{max}(k), \\ \mathbf{f}_{min}(k) &\leq \mathbf{f}(k) \leq \mathbf{f}_{max}(k), \end{aligned} \quad (5)$$

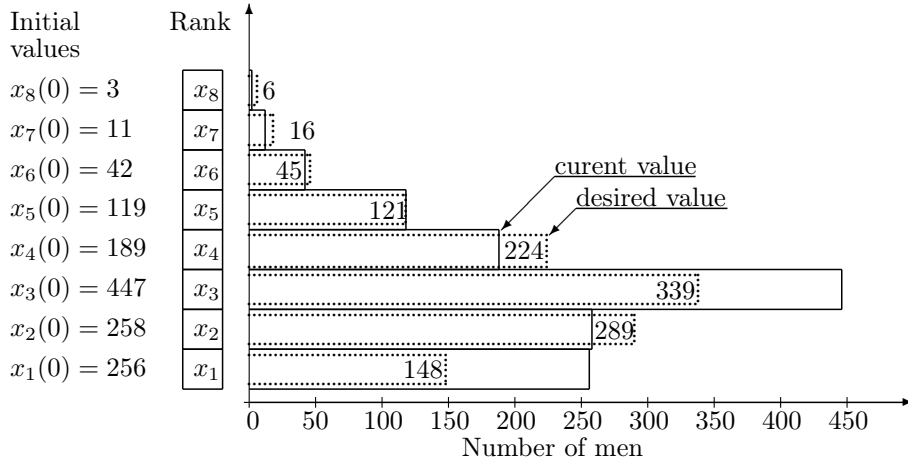


Fig. 2. Example of the current number of humans at a particular rank and the desired number of humans for eight ranks

where $\mathbf{u}_{min}(k)$ and $\mathbf{u}_{max}(k)$ are vectors of the lower and upper boundaries for recruitment, $\mathbf{r}_{min}(k)$ and $\mathbf{r}_{max}(k)$ are vectors of the lower and upper boundaries for transitions between ranks and $\mathbf{f}_{min}(k)$ and $\mathbf{f}_{max}(k)$ are vectors of the lower and upper boundaries for fluctuations. All boundaries are time dependent which increases the complexity of the addressed optimization problem.

3. Description of the optimization technique

The genetic algorithm for finding the optimal strategy for every rank was applied to the optimization. That means that we encoded all the coefficient values $\mathbf{r}(k)$ and $\mathbf{f}(k)$, as well as the input to the system \mathbf{u} , into a binary string for all the following time steps. The total dimensionality of the resulting optimization problem was $k \times t$, where k is the number of ranks, and t is the number of time steps, for example, for 8 ranks and 8 time steps the dimensionality was 64. For every variable the minimum and maximum values were set, and the algorithm worked within these borders.

We also used the self-configuring procedure in our genetic algorithm which allows an automatic selection of the most suitable set of genetic operators during the algorithm run. This self-configuring procedure is important as the efficiency of the genetic algorithm strongly depends on the types of genetic operators used, and the optimal combination of operators cannot be determined beforehand.

The applied self-configuration method is based on the idea of encouraging those operators which received the highest total fitness in the current generation [8,9]. This approach has proved its efficiency in the solving of hard real-world optimization problems [10,11] and has been recommended for practical use.

Let z be the number of different operators of the i -th type. The starting probability values are set to $p_i = 1/z$. The success estimation for every type of operator is performed based on the averaged fitness values:

$$AvgFit_i = \frac{\sum_{j=1}^{n_i} f_{ij}}{\sum_{j=1}^{n_i} 1}, \quad i = 1, 2, \dots, z, \quad (6)$$

where n_i is the number of offspring formed with the i -th operator, f_{ij} is the fitness value of the j -th offspring, obtained with i -th operator and $AvgFit_i$ is the average fitness of the solutions, obtained with i -th operator.

Then the probability of applying the operator, whose $AvgFit_i$ value is the highest among all the operators of this type, is increased by $(z \cdot K - K)/(z \cdot N)$, and the probabilities of applying other operators are decreased by $K/(z \cdot N)$, where N is the number of evolutionary algorithm generations and K is a constant equal to 0.5.

4. Application of the finite automaton

In order to eliminate undesired oscillations in the transitions and fluctuations, the finite automaton A was developed [12], which considers strategies with one extremum point where the set of automaton states is $S = \{S_0, S_1, S_2, S_3, S_4, S_5\}$, the comparison alphabet is $A = l, e, g$, the initial state is $i = S_0$ and the set of terminal states is $T = \{S_0, S_1, S_2, S_3, S_4\}$.

The transition function of A , $\delta : S \times A \rightarrow S$, is presented by the Fig. 3. Optimization is augmented by finite automaton A :

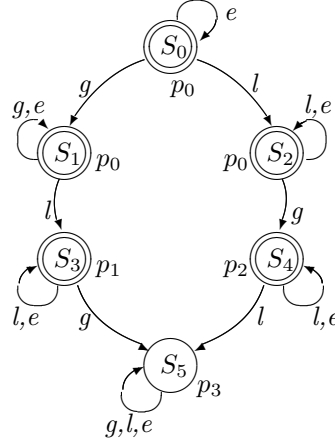


Fig. 3. Scheme of a finite automaton used to track the oscillations

$$J = \min_{u,r,f} (\mathbf{A}[\mathbf{r}, \mathbf{f}, \sum_{k=1}^{t_q} (\mathbf{z}(k) - \mathbf{x}(k))^T (\mathbf{z}(k) - \mathbf{x}(k))]) \quad (7)$$

subject to:

$$\begin{aligned} \mathbf{u}_{min}(k) &\leq \mathbf{u}(k) \leq \mathbf{u}_{max}(k), \\ \mathbf{r}_{min}(k) &\leq \mathbf{r}(k) \leq \mathbf{r}_{max}(k), \\ \mathbf{f}_{min}(k) &\leq \mathbf{f}(k) \leq \mathbf{f}_{max}(k), \end{aligned} \quad (8)$$

where \mathbf{A} represents the applied automaton. This is applied to the evolutionary algorithm and alters the value of the minimization function when the terminal state is not acceptable according to the appropriate weight, which eliminates improper strategies. The automaton is also defined

by the penalty coefficients p_0 , p_1 , p_2 and p_3 according to the number of alternating steps that were exercised by a particular strategy.

5. Constraint handling methods

To track infeasible solutions which contain oscillations of input parameters, such as $r(k)$ and $f(k)$, a penalty method was developed. The basic idea behind this method is that the discrete derivative of each r and f can be used to see if this function changed its orientation, i.e. the function is non-monotonic.

In the genetic algorithm, after evaluating the error, the feasibility is checked. To calculate the penalty value, we calculated the number of times when the derivative was more than or equal to zero, and the number of times when it was less than zero. Also, in each case we calculated the sum of the derivative values for positive and negative points separately. Besides, the derivative value for the input to the system \mathbf{u} was normalized to the interval $[0,1]$, as its value is much bigger than for the rest of variables.

After this, two conditions were checked. Firstly, if all derivative values are positive or negative, the penalty value for this time series is zero. Secondly, if there were several positive and several negative values, than the penalty size was set to the smallest module value between the two sums of positive and negative derivatives respectively. The penalty calculation can also be described as follows:

$$penalty = \begin{cases} 0, & \text{if } (N_p = T - 1 \text{ or } N_n = T - 1), \\ |S_n|, & \text{if } S_p < S_n, \\ |S_p|, & \text{if } S_n < S_p, \end{cases} \quad (9)$$

where N_p and N_n are the numbers of positive and negative derivative values, and S_p and S_n are the sums of the positive and negative derivative values respectively. These heuristic penalty values are calculated based on the idea that if most of the time the derivative was higher than zero, than the negative values should be changed to positive ones, so that these time series would become feasible, and vice versa.

In the genetic algorithm we used a modification of the dynamic penalties approach, so that the overall penalty value for all the time series was:

$$TotalPenalty = C \cdot \left(\sum_{i=0}^{k-1} penalty_r + \sum_{i=0}^k penalty_f + penalty_u \right) \cdot \sqrt{G}, \quad (10)$$

where C is a penalty weight constant and G is the current generation number, so that the penalty size increases at each generation.

We also used a modification of the finite automaton shown in figure 3 as a constraint handling method, and its main idea was that if the system ends up in states S_3 , S_4 or S_5 , it means that the corresponding coefficient made at least one oscillation, which is not desirable. So, in this case the finite automaton returned some value showing that this time series is not desirable and forming the penalty function. The return value is shown as the p_i value, defined by the equation:

$$penalty = \begin{cases} 0, & \text{if } p_0, \\ c_{i+1} - c_i, & \text{if } p_1, \\ c_i - c_{i+1}, & \text{if } p_2, \\ |c_{i+1} - c_i| + penalty_{i-1}, & \text{if } p_3, \end{cases} \quad (11)$$

where c_i is the coefficient value for the state i , and $penalty_{i-1}$ is the penalty value for the previous state. The return value is zero in states S_0 , S_1 and S_3 . For states S_3 and S_4 the return value depends on the size of the oscillation tracked, and for the final state S_5 the return value was the sum of the current and previous penalty during the oscillations of the time series.

6. Experimental results

For testing different combinations of the approaches discussed previously, for the genetic algorithm we used the following parameters:

- number of individuals: 1000,
- number of generations: 10000,
- tournament size: 25,
- penalty weights: 40, 4000 and 40000;
- the selection, crossover and mutation types were self-configuring.

First, we solved the problem of finding the optimal values of \mathbf{r} , \mathbf{f} and \mathbf{u} as an unconstrained optimization problem. In this case, we always observed huge oscillations for these parameters for every run of the algorithm performed. Fig. 4 shows the resulting values of \mathbf{r} and \mathbf{f} . The goal function mean square error in this case is 2.540.

The second series of experiments was performed using the finite automata to check the feasibility and calculate the penalty values. The penalty weight was set to 40. In this case we were able to obtain significantly fewer oscillations, although they are still presented. Fig. 5 depicts the results. The goal function mean square error in this case is 2.547.

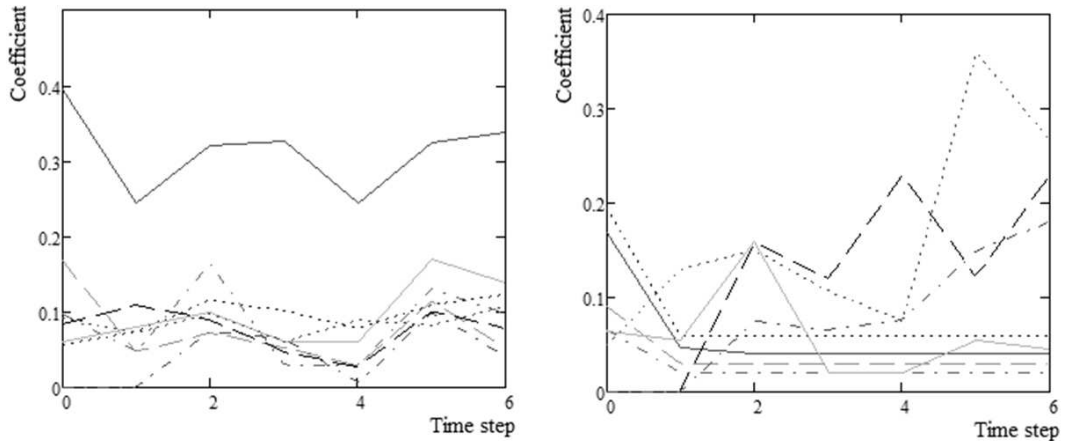


Fig. 4. Oscillations of transition and fluctuation coefficients (r 's and f 's) in unconstrained case

After obtaining the results with the standard finite automaton, we applied our modification with different return values depending on where the calculation stopped. The weight coefficient for this case was set to 40000. Fig. 6 depicts the results, and the mean square error for this case is equal to 2.684.

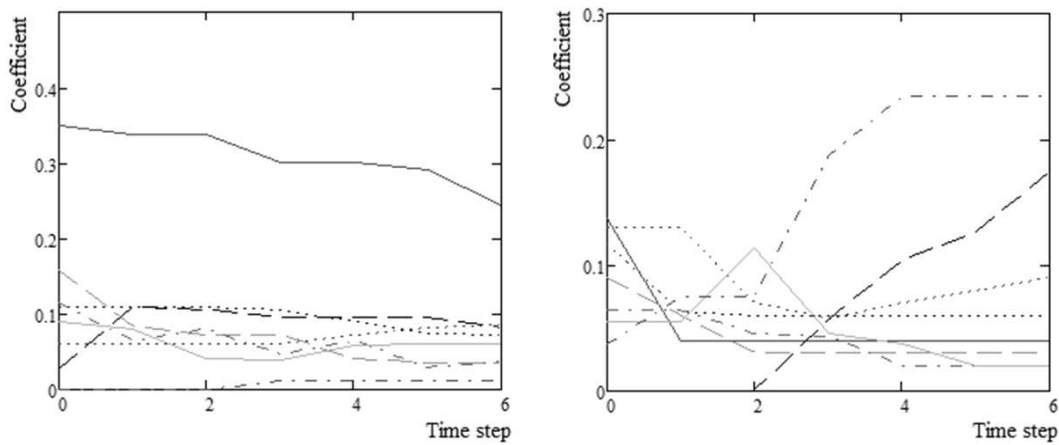


Fig. 5. Oscillations of transition and fluctuation coefficients (r 's and f 's) when using the finite automaton

In the following set of experiments we applied the derivative-based penalty assignment approach, and the penalty weight value of 4000. In this case there are still small oscillations, although they are not seen in Fig. 7. The goal function mean square error in this case is 2.580.

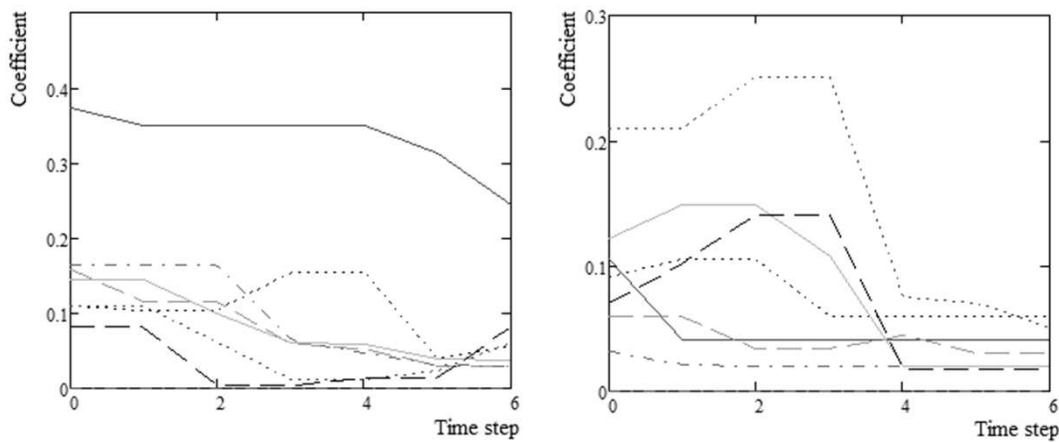


Fig. 6. Oscillations of transition and fluctuation coefficients (r 's and f 's) when using the modified finite automaton

Conclusion

The experimental results show that we managed to solve the manpower management problem by modelling the process and representing it as an optimization problem. By applying the self-configuring genetic algorithm we were able to solve this complex problem with 64 variables for a real example and receive satisfactory results. The main disadvantage of this result was the oscillations that we observed in the change in transition coefficients. These oscillations

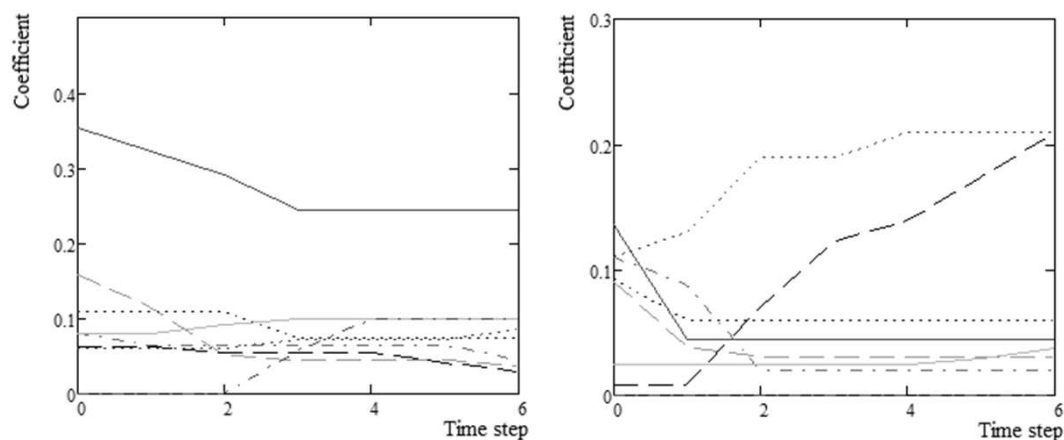


Fig. 7. Oscillations of transition and fluctuation coefficients (r 's and f 's) when using the derivative-based method

are not desirable, as they mean that the system policy must be significantly changed during the system operation which may bring additional costs in real-life applications. By adding constraint handling methods, firstly the finite automaton then its modification and finally the derivative-based method, we were able to receive non-oscillatory behaviour for the transition coefficients without a significant reduction in achieving the goal state of the system. The finite automaton gave much better results compared to the unconstrained case, although there were still some oscillations for several transition coefficients. The derivative-based method allowed even better results to be achieved, so that we received almost no oscillations in the final solution. The described methods and approaches were successfully applied to the human resource management and could also be applied for solving similar supply chain problems. The approach proposed could also be applied as the general improvement of a control principle where oscillations might occur, although for real-time application hardware realization would be needed, e.g. FPGA.

This research is financed by Slovenian Research Agency ARRS, Proj. No.: BI-RU/14-15-047 and Research Program Group P5-0018 (A).

References

- [1] A.Škraba, M.Kljajić, P.Papler, D.Kofjač, M.Obed, Determination of recruitment and transition strategies, *Kybernetes*, **40**(2011), no. 9/10, 1503–1522.
- [2] A.Mehlman, An approach to optimal recruitment and transition strategies for manpower systems using dynamic programming, *Journal of Operational Research Society*, **31**(1980), no. 11, 1009–1015.
- [3] R.Pastor, J.Olivella, Selecting and adapting weekly work schedules with working time accounts. A case of a retail clothing chain, *European Journal of Operational Research*, **184**(2008), no. 1, 1–12.
- [4] J.W.Forrester, *Industrial Dynamics*, MIT Press, Cambridge, MA, (1973).

- [5] T.Kok, F.Janssen, J.Doremalen, E.Wachem, M.Clerkx, W.Peeters, Philips Electronics Synchronizes its Supply Chain to End the Bullwhip Effect, *Interfaces*, **35**(2005), no. 1, 37–48.
- [6] C.D.Tarantilis, Editorial: Topics in Real-time Supply Chain Management, *Computers & Operations Research*, (2008), no. 35, 3393–3396.
- [7] E.S.Bres, D.Burns, A.Charnes, W. W.Cooper, A Goal Programming model for planning officer accessions, *Management Science*, **26**(1980), no. 8, 773–783.
- [8] E.Semenkin, M.Semenkina, Self-configuring Genetic Algorithm with Modified Uniform Crossover Operator, Y. Tan, Y. Shi, and Z. Ji (Eds.): *Advances in Swarm Intelligence*, Lecture Notes in Computer Science 7331, Springer-Verlag Berlin Heidelberg, 2012, 414–421.
- [9] E.Semenkin, M.Semenkina, Self-Configuring Genetic Programming Algorithm with Modified Uniform Crossover, Proceedings of the Congress on Evolutionary Computations of the IEEE World Congress on Computational Intelligence (CEC WCCI 2012), Brisbane, Australia, 2012, 1918–1923.
- [10] E.Semenkin, M.Semenkina, The Choice of Spacecrafts' Control Systems Effective Variants with Self-Configuring Genetic Algorithm, J.L.Ferrier et. al. (Eds.), *Informatics in Control. Automation and Robotics*, Proceedings of the 9th International Conference ICINCO, v. 1, 2012, 84–93.
- [11] E.Semenkin, M.Semenkina, Stochastic Models and Optimization Algorithms for Decision Support in Spacecraft Control Systems Preliminary Design, J.-L. Ferrier et al. (eds.), *Informatics in Control, Automation and Robotics*, Lecture Notes in Electrical Engineering 283, Springer International Publishing Switzerland, 2014, 51-65.
- [12] A.Škraba, E.Semenkin, M.Semenkina, D.Kofjač, A.Žnidaršič, Č.Rozman, M.Maletič, Sh.Akhmedova, V.Stanovov, Development of discrete manpower model and determination of optimal control strategies, Proceedings of the XVIII International Scientific Conference "Reshetnev Readings", 2014, 421–423.

Применение самоконфигурируемого генетического алгоритма для управления человеческими ресурсами

**Андрей Шкраба, Даворин Кофьяч
Аня Жнидаршич, Матьяж Малетич
Чртомир Розман, Евгений С. Семенкин
Мария Е. Семенкина, Владимир В. Становов**

В данной статье описывается задача управления человеческими ресурсами, которая может быть актуальной для организаций в период реструктуризации. Поведение системы описывается динамической имитационной моделью, аналогичной модели цепи поставок с несколькими рангами. Поиск оптимальной комбинации передаточных коэффициентов, включающих также коэффициенты флуктуаций, сводится к задаче оптимизации. Для решения этой задачи применяется самоконфигурируемый генетический алгоритм с несколькими методами учета ограничений. Ограничения в данной проблеме возникают в связи с необходимостью избежать нежелательных осцилляций в системе. Результаты показывают, что поставленная задача может быть эффективно решена предложенными методами.

Ключевые слова: человеческие ресурсы, имитационное моделирование, генетический алгоритм, самоконфигурирование, условная оптимизация.