Proceedings of the World Congress on Engineering and Computer Science 2014 Vol II
WCECS 2014, 22-24 October, 2014, San Francisco, USA

Simulated Metamorphosis - A Novel Optimizer

Michael Mutingi, Charles Mbohwa

***Abstract*—This paper presents a novel metaheuristic algorithm, simulated metamorphosis (SM), inspired by the biological concepts of metamorphosis evolution. The algorithm is motivated by the need for interactive, multi-objective, and fast optimization approaches to solving problems with fuzzy conflicting goals and constraints. The algorithm mimics the metamorphosis process, going through three phases: initialization, growth, and maturation. Initialization involves random but guided generation of a candidate solution. After initialization, the algorithm successively goes through two loops, that is, growth and maturation. Computational tests performed on benchmark problems in the literature show that, when compared to competing metaheuristic algorithms, SM is more efficient and effective, producing better solutions within reasonable computation times.**

***Index Terms*—Metamorphosis, evolution, optimization, algorithm, metaheuristics**

I. Introduction

SIMULATED Metamorphosis (SM) is a novel evolutionary approach to metaheuristic optimization inspired by the natural biological process of metamorphosis common in many insect species [1] [2]. The metaheuristic approach is motivated by several problem situations in the operations research and operations management community, such as nurse scheduling [3] [4] [5], vehicle routing problems [6], and task assignment [7]. In particular, the metaheuristic is motivated by hard optimization problems that are associated with multiple conflicting objectives, imprecise fuzzy goals and constraints, and the need for interactive optimization approaches that can incorporate the choices, intuitions and expert judgments of the decision maker [4] [8].

In a fuzzy environment, addressing hard optimization problems with conflicting goals requires interactive tools that are fast, flexible, and easily adaptable to specific problem situations. Decision makers often desire to use judicious approaches that can find a cautious tradeoff between the many goals, which is a common scenario in real world problems [2]. Addressing ambiguity, imprecision, and uncertainties of management goals is highly desirable in practice [4] [8]. For instance, in a hospital setting, where nurses are often allowed to express their preferences on shift schedules, the decision maker has to incorporate the imprecision in preferences and management goals and choices. Moreover, it is important to balance workload assignment, if shift fairness and equity are to be achieved. Preferences of patients or clients have to be considered as well. Though imprecise and conflicting, these factors have to be considered when constructing work schedules [4] [5]. Similar situations are commonplace in hard combinatorial problems.

In view of the above highlighted needs for interactive fuzzy multi-objective optimization approaches, the purpose of this research is to introduce a novel simulated metamorphosis algorithm, a fuzzy metaheuristic algorithm that is derived from the biological metamorphosis evolution process. Our objectives are as follows:

1) To present the basic concepts of the metamorphosis evolution process;
2) To derive, from the metamorphosis concepts, an interactive fuzzy evolutionary algorithm; and,
3) To apply the algorithm to typical nurse scheduling problems, demonstrating its effectiveness.

The rest of the paper is structured as follows. The next section presents the basic concepts of metamorphosis evolution. Section III proposes the simulated metamorphosis algorithm. Section IV presents the nurse scheduling problem. Section V presents a simulated metamorphosis for the nurse scheduling problem. Computational illustrations are provided in Section VI. Section VII concludes the paper.

II. Metamorphosis: Basic Concepts

Metamorphosis is an evolutionary process common in insects such as butterflies [2]. The process begins with an egg that hatches into an instar larva (instar). Subsequently, the first instar transforms into several instar larvae, then into a pupa, and finally into the adult insect [1] [2]. The process is uniquely characterized with radical evolution and hormone controlled growth and maturation.

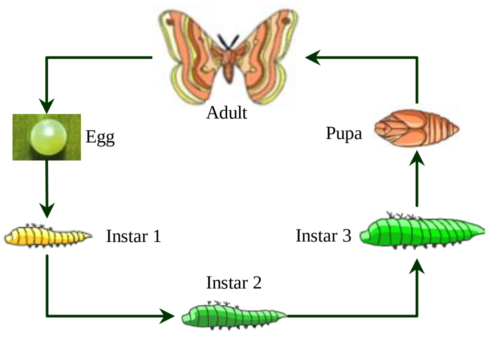

Fig. 1 Metamorphosis evolution

Manuscript received June 13, 2014; revised August 30, 2014.

M. Mutingi is a doctoral student with the Faculty of Engineering and the Built Environment, University of Johannesburg, Bunting Road Campus, P. O. Box 524, Auckland Park 2006, Johannesburg, South Africa (phone: +27789578693; e-mail: mmutingi@gmail.com).

C. Mbohwa is a professor with the Department of Quality and Operations Management, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg, Bunting Road Campus, P. O. Box 524, Auckland Park 2006, Johannesburg, South Africa (e-mail: cmbohwa@uj.ac.za).

A. Metamorphosis Evolution

When an insect grows and develops, it must periodically shed its rigid exoskeleton in a process called molting. The insect grows a new loose exoskeleton that provides the insect with room for more growth [2]. The insect transforms in body structure as it molts from a juvenile to an adult form, a process called metamorphosis.

The concept of metamorphosis refers to the change of physical form, structure, or substance; a marked and more or less abrupt developmental change in the form or structure of an animal (such as a butterfly or a frog) occurring subsequent to hatching or birth [1]. A species changes body shape and structure at a particular point in its life cycle, such as when a tadpole turns into a frog. Sometimes, in locusts for example, the juvenile form is quite similar to the adult one. In others, they are radically different, and unrecognizable as the same species. The different forms may even entail a completely new lifestyle or habitat, such as when a ground-bound, leaf-eating caterpillar turns into a long distance flying, nectar-eating butterfly.

A. Hormonal Control

Insect molting and development is controlled by several hormones [1]. The hormones trigger the insect to shed its exoskeleton and, at the same time, grow from smaller juvenile forms (e.g., a young caterpillar) to larger adult forms (e.g., a winged moth) [2]. The hormone that causes an insect to molt is called ecdysone. The hormone, in combination with another, called juvenile hormone, also determines whether the insect will undergo metamorphosis.

III. Simulated Metamorphosis

There are three basic phases: initialization, growth, and maturation. Each of these phases has specific operators.

A. Initialization Phase

In the initialization stage, an initial solution is created as a seed for the evolutionary algorithm. In our approach, we use a problem specific heuristic that is guided by hard constraints of the problem. This ensures generation of a feasible initial solution. Alternatively, a decision maker can enter a user-generated solution as a seed. The initial candidate solution s_t (t = 1,…,T) consists of constituent elements e_i (i = 1,…,I) where I is the constituent number of elements in the candidate solution.

Following the creation phase, the algorithm goes into a loop for a maximum of T iterations (generations).

B. Growth Phase

The growth phase comprises the *evaluation*, *transformation*, and the *regeneration* operators.

1) Evaluation

The choice of the evaluation function is very crucial to the success of evaluation operator and the overall algorithm. First, the evaluation function should ensure that it measures the relevant quality of the candidate solution. Second, the function should capture the actual problem characteristics, particularly the imprecise, conflicting and multi-objective nature of the goals and constraints. Third, the fitness function should be easy to evaluate and compute.

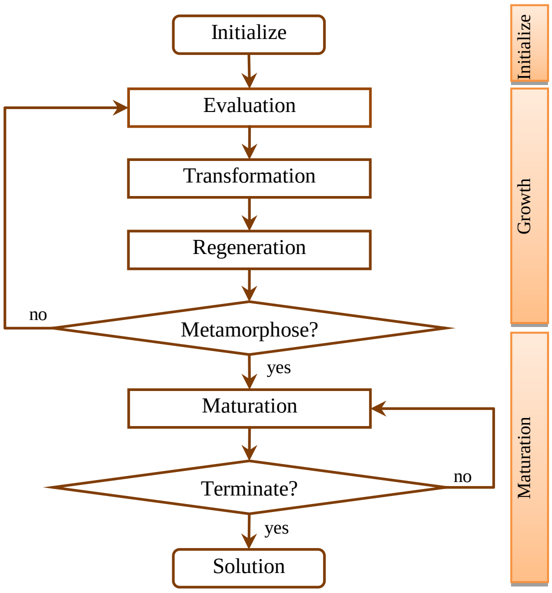

Fig. 2 Simulated metamorphosis algorithm

The evaluation function F_t, at iteration t, should be a normalized function obtained from its constituent normalized functions denoted by μ_h (h = 1,…,n), where n is the number of constituent objective functions.

In this approach, we use fuzzy multi-factor evaluation method, that is,

$$F_t(s_t) = \sum_h w_h \mu_h(s_t) \tag{1}$$

where, s_t is the current solution at iteration t; and w_h denotes the weight of the function μ_h. The use of the max-min operator is avoided so as to prevent possible loss of vital information.

2) Transformation

The growth mechanism is achieved through selection and transformation operators. Selection determines whether a constituent element e_i of the candidate solution s_t should be retained for the next iteration, or selected for transformation operation. The goodness or fitness η_i of element e_i (i = 1,…,I) is compared with probability $p_t \in [0,1]$, generated at each iteration t. That is, if $\eta_i \leq p_t$, then e_i is transformed, otherwise, it will survive into the next iteration. Deriving from the biological metamorphosis, the magnitude of p_t should decrease over time to guarantee convergence. From our preliminary empirical computations, p_t should follow a decay function of the form,

$$p_t = p_0 e^{-at/T} \tag{2}$$

where, $p_0 \in [0,1]$ is a randomly generated number; T is the maximum number of iterations; a is an adjustment factor.

It follows that the higher the goodness, the higher the likelihood of survival in the current solution. Therefore, elements with low goodness are subjected to growth. The magnitude of p_t controls the growth rate, which emulates the inhibition or juvenile hormone.

To avoid loss of performing elements, new elements are

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

compared with the rejected ones, keeping the better ones. A pre-determined number of rejected elements are kept in the reject list *R* for future use in the regeneration stage.

3) Regeneration

The regeneration operator has a repair mechanism that considers the feasibility of the candidate solution. All infeasible elements are repaired using problem domain specific heuristics, developed from problem constraints. Elements in the reject list *R* are used as food for enhancing the repair mechanism.

After regeneration, the candidate solution is tested for readiness for transition to the maturation phase. This is controlled by the dissatisfaction level (juvenile hormone) m_t at iteration *t*, represented by the expression,

$$m_t = 1 - \mu_1 \wedge \mu_2 \wedge ... \wedge \mu_n \tag{3}$$

Here, $\mu_1,\ldots,\mu_n$, represent the satisfaction level of the respective objective functions; "∧" is the min operator. This implies that the growth phase repeats until a pre-defined acceptable dissatisfaction m_0 is reached. However, if there is no significant change in m_t after a pre-defined number of trials, then the algorithm proceeds to the maturation phase.

C. Maturation

The maturation phase is a loop consisting of *intensification* and *post-processing* operators. The aim is to bring to maturity the candidate solution, so as to obtain the best solution.

1) Intensification

The aim of the intensification operator is to ensure complete search of an improved solution in the neighborhood of the current solution. This helps to improve the current solution further. Howbeit, at this stage, the juvenile hormone has ceased to control or balance the growth of the solution according to the constituent fitness functions.

2) Post-processing

The post-processing operator is user-guided; it allows the user to interactively make expert changes to the candidate solution, and to re-run the intensification operator. As such, the termination of the maturation phase is user determined. This also ensures that expert knowledge and intuition are incorporated into the solution procedure. This enhances the interactive search power of the algorithm.

D. Comparing SM and Related Algorithms

The proposed SM algorithm has a number of advantages over related metaheuristics. Contrary to Simulated Annealing (SA) which makes purely random choices to decide the next move, SM employs intelligent selection operation to decide which changes to perform. Furthermore, SM takes advantage of multiple transformation operations on weak elements of the current solution, allowing for more distant changes between successive iterations.

The SM algorithm, like Genetic Algorithm (GA), uses the mechanics of evolution as it progresses from one generation to the other. GA necessarily keeps a number of candidate solutions in each generation as parents, generating offspring by a crossover operator. On the contrary, SM simulates metamorphosis, evolving a single solution under hormonal control. In addition, domain specific heuristics are employed to regenerate and repair the emerging candidate solution, developing it into an improved and complete solution. In retrospect, SM reduces the computation time needed to maintain a large population of candidate solutions in GA.

The selection process in the SM is quite different from GA and other related evolutionary algorithms. While GA uses probabilistic selection to retain a set of good solutions from a population of candidate solutions, SM selects and discards inferior elements of a candidate solution, according to the goodness of each element. This enhances the computational speed of the SM procedure.

At the end of the growth phase, the SM algorithm goes through maturation phase where intensive search process is performed to refine the solution, and possibly obtain an improved final solution. The algorithm allows the decision maker to input his/her managerial choices to guide the search process. This interactive facility gives SM an added advantage over other heuristics.

The proposed algorithm uses hormonal control to enhance and guide its global multi-objective optimization process. This significantly eliminates unnecessary search through regions with inferior solutions, hence, improving the search efficiency of the algorithm. In summary, the above mentioned advantages provide the SM algorithm enhanced convergence characteristics that enable the algorithm to perform fewer computations relative to other evolutionary algorithms.

IV. The Nurse Scheduling Problem

The nurse scheduling problem (NSP) is a hard multi-criteria optimization problem that involves assignment of different types of shifts and off days to nurses over a period of up to one month. The decision maker considers a number of conflicting objectives, choices, and preferences associated with the healthcare organization and individual nurses [9] [10] [11]. In practices, contractual work agreements govern the number of assignable shifts and off days per week. Imprecise personal preferences should be satisfied as much as possible. Typically nurses are entitled to day shift *d*, night shift *n*, and late night shift *l*, with holidays or days-off *o* [12]. Table I lists and describes common shift types and their time allocations.

The primary aim is to search for a schedule that satisfies a given set of hard constraints while minimizing a specific cost function [10] [12]. However, in practice, individual nurse preferences, which are often imprecise, have to be satisfied to the highest degree possible; the higher the degree of satisfaction, the higher the schedule quality [9]. This ensures not only healthcare service quality, but also satisfactory healthcare work environment (job satisfaction).

TABLE I
TYPICAL SHIFT TYPES

Shift	Shift Description	Time allocation
1	*d*: day shift	0800 - 1600 hrs
2	*e*: night shift	1600 - 2400 hrs
3	*n*: late night shift	0000 - 0800 hrs
4	*o*: off days as nurse preferences	

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
TYPICAL CONSTRAINTS TYPES

Constraints	Description of the constraint
Daily Restrictions	C1: Assign each nurse at most one shift per day. C2: Shift sequences (*e-d*), (*n-e*) and (*n-d*) are not permissible. C3: Assigned legal holidays = Legal holidays. C4: Interval between night shifts should ≥ 1 week.
Nurse Preferences	P1: Preferred or desired day off or holidays. P2: Fairness or equality of shifts for each nursing staff P3: Congeniality - compatible shift assignments between work mates

Table II provides a list of typical hard constraints (C1 to C4) and soft constraints (P1 to P3). In most cases, hard constraints consist of daily restrictions that arise from legislative laws, while soft constraints arise from nurse preferences [8] [9] [10].

V. SIMULATED METAMORPHOSIS FOR NURSE SCHEDULING

In this section, we present an application of simulated metamorphosis for nurse scheduling in a fuzzy environment with multiple objectives.

A. *Initialization*

The initialization algorithm is designed such that, while assigning shifts at random, all hard constraints are satisfied. This is achieved by incorporating all the hard constraints into the initialization procedure. In addition, the coding schema ensures that only one shift is assigned to a nurse on each day, thus satisfying constraint C1. This improves the speed of the initialization process. Fig. 3 presents an enhanced initialization algorithm that incorporates hard constraints.

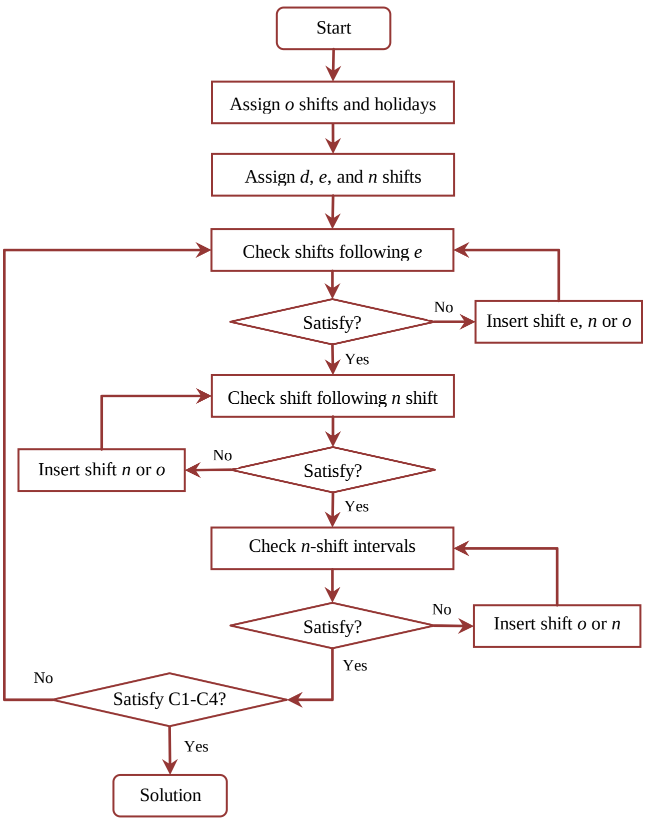

Fig. 3 SM initialization procedure incorporating hard constraints

B. *Growth Phase*

1) *Evaluation*

The goodness, fitness, or quality of a solution is a function of how much it satisfies soft constraints. As such, fitness is a function of the weighted sum of the satisfaction of soft constraints. Thus, each soft constraint is represented as a normalized fuzzy membership function in [0,1]. In this study, we use two types of membership functions: (a) triangular functions, and (b) interval-valued functions, as show in Fig. 4.

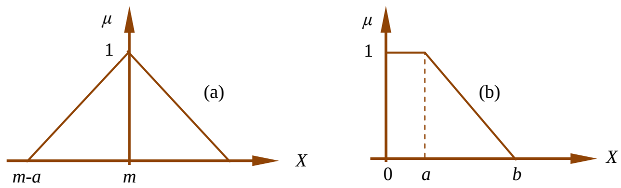

Fig. 4 Linear membership functions

In (a), the satisfaction level is represented by a fuzzy number *A*<*m*,*a*>, where *m* denotes the centre of the fuzzy parameter with width *a*. Thus, the corresponding membership function is,

$$\mu_A(x)=\begin{cases}1-\dfrac{|m-x|}{a} & \text{If } m-a\le x\le m+a\\ 0 & \text{If otherwise}\end{cases} \tag{4}$$

In (b), the satisfaction level is represented by a decreasing linear function where [0,*a*] is the most desirable range, and *b* is the maximum acceptable. Therefore, the corresponding function is,

$$\mu_B(x)=\begin{cases}1 & \text{If } x\le a\\ (b-x)/(b-a) & \text{If } a\le x\le b\\ 0 & \text{If otherwise}\end{cases} \tag{5}$$

Membership Function 1 - Workload Variation: For fair workload assignment, the workload h_i for each nurse *i* should be as close as possible to the mean workload *w*. Therefore, the workload variation $x_i=h_i-w$ should be minimized. Assuming symmetrical triangular membership function from (3), we obtain,

$$\mu_1(x_i)=\mu_A(x_i) \tag{6}$$

where, x_i is workload variation for nurse *i* from mean *w* of the fuzzy parameter, with width *a*.

Membership Function 2 - Allocated Days Off: This membership function measures the variation of the allocated days off from the mean. We assume symmetrical triangular membership function derived from (3) as follows;

$$\mu_2(x_i)=\mu_A(x_i) \tag{7}$$

where, x_i is the actual variation of days off for nurse *i* from the mean *m* of the fuzzy parameter with width *a*.

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Membership Function 3 - Variation Night Shifts: For shift fairness the variation x_i of the number of night shifts (shifts e and n) allocated to each nurse i should be as close as possible to the mean allocation m. Assuming symmetrical triangular membership function from (3), we obtain,

$$\mu_3(x_i) = \mu_A(x_i) \tag{8}$$

where, x_i is the variation of number of nights shifts allocated to nurse i from mean m of the fuzzy parameter, with width a.

Membership Function 4 - Congeniality: This membership function measures the compatibility (congeniality) of staff allocated similar shifts; the higher the congenialities, the higher the schedule quality. In practice, a decision maker sets limits to acceptable number of uncongenial shifts x_i for each nurse i to reflect satisfaction level. Assuming interval-valued functions in Fig. 4 (b), the corresponding membership function is,

$$\mu_4(x_i) = \mu_B(x_i) \tag{9}$$

where, x_i is the actual number of uncongenial allocations; a is the upper limit to the preferred uncongenial shifts; b is the maximum uncongenial shifts.

Membership Function 5 – Understaffing: High quality schedule minimize as much as possible the understaffing for each shift k. In practice, the level of understaffing $x_j = \sum u_k$ in each day j should be within acceptable limits. This can be represented by a linear interval-valued membership function derived from (4);

$$\mu_5(x_j) = \mu_B(x_j) \tag{10}$$

where, x_j is the staffing variation from mean m of the fuzzy parameter, with width a.

Membership Function 6 – Overstaffing: For high quality schedule, overstaffing o_k for each shift k should be minimized as much as possible. In a practical setting, the level of overstaffing $x_j = \sum o_k$ for all shifts in each day j should be within acceptable limits, which can be represented by a linear interval-valued membership from (4);

$$\mu_6(x_j) = \mu_B(x_j) \tag{11}$$

where, x_j is the staffing variation from mean m of the fuzzy parameter, with width a.

The Overall Fitness: The fitness for each shift pattern i for each nurse is obtained from the weighted sum of the first four membership functions. For horizontal fitness As such, the fitness for each shift pattern (or element) i is obtained according to the following expression;

$$\eta_i = \sum_{z=1}^{4} w_z \mu_z(x_i) \qquad \forall i \tag{12}$$

where, w_z is the weight of each function μ_z, such that condition $\sum w_z = 1.0$ is satisfied.

Similarly, the fitness according to shift requirement in each day j is given by,

$$\lambda_j = \sum_{z=5}^{6} w_z \mu_z(x_j) \qquad \forall j \tag{13}$$

where, w_z is the weight of each function μ_z, with $\sum w_z = 1.0$.

The overall fitness of the candidate solution is given by the expression,

$$f = \left(\frac{\eta}{\omega_1} \wedge 1\right) \wedge \left(\frac{\lambda}{\omega_2} \wedge 1\right) \tag{14}$$

where, $\lambda = \lambda_1 \wedge \dots \wedge \lambda_4$; $\mu = \mu_1 \wedge \mu_2$; ω_1 and ω_2 are the weights associated with η and λ, respectively; "$\wedge$" is the min operator.

The weights w_z, ω_1 and ω_2 offer the decision maker an opportunity to incorporate his/her choices reflecting expert opinion and preferences of the management and the nurses. This feature gives the SM algorithm an added advantage over other methods.

2) Transformation

In NSP, elements are two-fold: one that represents horizontal shift patterns, denoted by e_i, and another representing the vertical shift allocations for each day, denoted by e_j. Fitness η_i and λ_j of each element are probabilistically tested for transformation by comparing with a random number $p_t \in [0,1]$, generated at each iteration t. A transformation probability $p_t = p_0 e^{-t/T}$ is used to probabilistically change elements e_i and e_j using column-wise and row-wise heuristics to improve the solution.

3) Regeneration

Regeneration repairs infeasible elements using a mechanism similar to the initialization algorithm which incorporates hard constraints. Based on the juvenile hormone level m_t at iteration t, the candidate solution is then tested for readiness for maturation,

$$m_t = 1 - (\eta_1 \wedge \eta_2 \wedge \eta_3 \wedge \eta_4) \wedge (\lambda_1 \wedge \lambda_2) \tag{15}$$

The growth phase repeats until a pre-defined acceptable dissatisfaction m_0 is reached. However, the algorithm proceeds to the maturation phase if there is no significant change ε in m_t within a predetermined number of iterations, with the value of ε set in the order of 10^{-6}.

C. Maturation

Intensification ensures complete search of a near-optimal solution in the neighbourhood of the current solution. In the post-processing stage the user interactively makes expert changes to the candidate solution, and to execute intensification. Expert knowledge and intuition are coded in form of possible adjustments through weights $w_1,\dots,w_4$ and ω_1, ω_1. Illustrative computations are presented in the next section.

VI. Computational Results and Discussion

To illustrate the effectiveness of the proposed SM algorithm, computational experiments were carried out on a typical nurse scheduling problem with 13 nurses over a a planning horizon of 14 days.

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Fitness η_i
Nurse 1	*o*	*o*	*e*	*e*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	*e*	*e*	0.6308
Nurse 2	*e*	*n*	*o*	*d*	*d*	*d*	*d*	*o*	*d*	*d*	*e*	*e*	*o*	*d*	0.7308
Nurse 3	*d*	*d*	*d*	*d*	*d*	*n*	*n*	*n*	*n*	*o*	*d*	*d*	*o*	*d*	0.8635
Nurse 4	*n*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*o*	*n*	*n*	*n*	0.8885
Nurse 5	*d*	*d*	*d*	*n*	*n*	*o*	*d*	*d*	*o*	*d*	*d*	*d*	*e*	*e*	0.8385
Nurse 6	*d*	*d*	*d*	*o*	*e*	*e*	*e*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	0.8385
Nurse 7	*d*	*d*	*o*	*d*	*n*	*n*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	0.7385
Nurse 8	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	*e*	*e*	*e*	*o*	*d*	*d*	0.8885
Nurse 9	*d*	*d*	*d*	*d*	*d*	*e*	*n*	*n*	*o*	*d*	*d*	*e*	*e*	*o*	0.8442
Nurse 10	*o*	*e*	*e*	*e*	*e*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	0.8385
Nurse 11	*d*	*d*	*d*	*d*	*d*	*o*	*e*	*e*	*e*	*n*	*o*	*d*	*d*	*o*	0.7308
Nurse 12	*e*	*n*	*n*	*n*	*o*	*d*	*d*	*d*	*n*	*n*	*o*	*o*	*d*	*d*	0.4865
Nurse 13	*d*	*n*	*n*	*o*	*d*	*d*	*o*	*d*	*d*	*o*	*d*	*n*	*n*	*n*	0.6615
Fitness λ_k	0.80	0.67	1.00	1.00	1.00	0.80	1.00	1.00	1.00	0.67	0.47	0.67	0.67	0.67	$f = 0.4667$

Fig. 5 Initial nurse schedule

	1	2	2	4	5	6	7	8	9	10	11	12	13	14	Fitness η_i
Nurse 1	*n*	*n*	*o*	*d*	*d*	*d*	*d*	*o*	*d*	*e*	*e*	*o*	*d*	*d*	0.6558
Nurse 2	*e*	*n*	*n*	*n*	*o*	*d*	*d*	*d*	*o*	*d*	*d*	*d*	*d*	*d*	0.9385
Nurse 3	*d*	*o*	*n*	*n*	*n*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	*e*	0.9385
Nurse 4	*n*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*o*	*n*	*n*	*n*	0.9385
Nurse 5	*d*	*d*	*d*	*d*	*d*	*n*	*n*	*n*	*n*	*o*	*d*	*d*	*o*	*d*	0.9385
Nurse 6	*o*	*d*	*d*	*d*	*e*	*e*	*e*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	0.8385
Nurse 7	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	*n*	*n*	*n*	*o*	*d*	*d*	0.9385
Nurse 8	*d*	*e*	*e*	*o*	*o*	*d*	*d*	*d*	*d*	*n*	*o*	*e*	*d*	*d*	0.7308
Nurse 9	*d*	*d*	*d*	*d*	*d*	*e*	*n*	*n*	*o*	*d*	*d*	*e*	*e*	*o*	0.8692
Nurse 10	*e*	*e*	*e*	*e*	*e*	*o*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	0.7942
Nurse 11	*d*	*d*	*d*	*d*	*d*	*d*	*e*	*e*	*e*	*e*	*n*	*d*	*o*	*o*	0.8692
Nurse 12	*d*	*d*	*o*	*e*	*n*	*n*	*o*	*d*	*d*	*d*	*d*	*d*	*d*	*e*	0.7885
Nurse 13	*d*	*d*	*d*	*o*	*d*	*d*	*d*	*d*	*e*	*o*	*e*	*n*	*n*	*n*	0.8692
Fitness λ_k	1.0	1.00	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	$f = 0.8197$

Fig. 6 Final nurse schedule

Fig. 5 shows the initial schedule created using the enhanced initialization procedure. The shift requirements for shifts *d*, *e*, and *n* are 7, 2, and 2, respectively. Only 6 out of 14 days have 100% satisfaction of shift requirements. Assume that, due to congeniality issues, nurse combinations (1,10) and (1,12) are to be avoided as much as possible. The fitness values for each shift pattern are obtained using expression (11). Similarly, the fitness values for each day are obtained from (12). The maximum number of iterations T = 200. The initial overall fitness is 0.4667, which is very low.

Fig. 6 shows the final nurse schedule obtained after 200 iterations. The solution shows a marked improvement in the fitness values of individual shift patterns. Also, there is a 100% satisfaction of the shift requirements in each day, which is a marked improvement from the initial solution. Consequently, the overall fitness value of the final schedule is 0.8197, which is a significant improvement from the initial schedule.

VII. Conclusions and Further Work

Motivated by the biological metamorphosis process and the need to solve multi-objective optimization problems with conflicting and fuzzy goals and constraints, this research proposed a simulated metamorphosis algorithm, based on the concepts of biological evolution in insects, including moths, butterflies, and beetles. The algorithm mimics the hormone controlled evolution process going through initialization, iterative growth loop, and finally maturation loop.

The suggested methods offers a practical approach to optimizing multi-objective problems with fuzzy conflicting goals and constraints such as the nurse scheduling, homecare nurse routing and scheduling, vehicle routing, job shop scheduling, and task assignment. Equipped with the facility to incorporate the user's choices and wishes, the algorithm offers an interactive approach that can accommodate the decision maker's expert intuition and experience, which is otherwise impossible with other optimization algorithms.

The proposed metaheuristic is efficient and effective. By using hormonal guidance and unique operators, the algorithm employs two successive iterative loops, working on a single candidate solution to efficiently search for the best solution.

Simulated metamorphosis is an invaluable addition to the operations research and operations management community, specifically to researchers concerned with multi-objective global optimization. Learning from the preliminary experimental tests of the algorithm, the application of the proposed approach can be extended to a number of practical hard problems such as task assignment, vehicle routing, home healthcare nurse scheduling, job sequencing, and time tabling.

Acknowledgment

The authors appreciate the reviewers for their invaluable comments on the previous version of this paper.

References

[1] G. Tufte, "Metamorphosis and Artificial Development: An Abstract Approach to Functionality," In G. Kampis, I. Karsai, and E. Szathm´ary (Eds.): ECAL 2009, Part I, LNCS 5777, Springer-Verlag Berlin Heidelberg, pp. 83–90, 2011.

[2] J.W. Truman and L.M. Riddiford, "Endocrine insights into the evolution of metamorphosis in insects," *The Annual Review of Entomolog*, vol. 47, pp. 467–500, 2002.

[3] A. Jan, M. Yamamoto, A. Ohuchi, "Evolutionary algorithms for nurse scheduling problem," *IEEE Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 196 – 203, July 2000.

[4] M. Mutingi, C. Mbohwa, "Healthcare staff scheduling in a fuzzy environment: A fuzzy genetic algorithm approach," *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management*, Bali, Indonesia, pp. 303-312, January 7 – 9, 2014.

[5] T. Inoue, T. Furuhashi, H. Maeda, and M. Takaba, "A proposal of combined method of evolutionary algorithm and heuristics for nurse scheduling support system," *IEEE Transactions on Industrial Electronics*, vol. 50, no.5, pp. 833-838, 2003.

[6] C.D. Tarantilis, C.T. Kiranoudis, V.S.A. Vassiliadis, "A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *European Journal of Operations Research*, vol. 152, pp. 148–158, 2004.

[7] M. Cheng, H.I. Ozaku, N. Kuwahara, K. Kogure, J. Ota, "Nursing care scheduling problem: Analysis of staffing levels," *IEEE Proceedings of the 2007 International Conference on Robotics and Biomimetics*, vol. 1, pp. 1715-1719, December 2007.

[8] S. Topaloglu, S. Selim, "Nurse scheduling using fuzzy modeling approach," *Fuzzy Sets and Systems*, vol. 161, pp. 1543–1563, 2010.

[9] M. Mutingi, C. Mbohwa, "A fuzzy genetic algorithm for healthcare staff scheduling," *International Conference on Law, Entrepreneurship and Industrial Engineering (ICLEIE'2013)*, pp. 239-243, April 15-16, 2013.

[10] S. Shaffer, "A rule-based expert system for automated staff scheduling. IEEE International Conference on Systems, Man, and Cybernetics, Decision Aiding for Complex Systems, vol.3 pp. 1691-1696, 1991.

[11] E. Burke, P. Causmaecker, V. G. Berghe, H. Landeghem, "The state of the art of nurse rostering," *Journal of Scheduling*, vol. 7, pp.441-499, 2004.

[12] B. Cheang, H. Li, A. Lim, and B. Rodrigues, "Nurse rostering problems–a bibliographic survey," *European Journal of Operational Research*, vol. 151, pp. 447–460, 2003.