Solving the Manufacturing Cell Design Problem via Invasive Weed Optimization

Ricardo Soto[1,2,3], Broderick Crawford[1,4,5], Carlos Castillo[1], and Fernando Paredes[6]

[1] Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
[2] Universidad Autónoma de Chile, Santiago, Chile
[3] Universidad Cientifica del Sur, Lima, Peru
[4] Universidad Central de Chile, Santiago, Chile
[5] Universidad San Sebastián, Santiago, Chile
[6] Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
{ricardo.soto,broderick.crawford}@ucv.cl
carlos.castillo.m@mail.pucv.cl,fernando.paredes@udp.cl

Abstract. Manufacturing plants are commonly organized in cells containing machines that process different parts of a given product. The Manufacturing Cell Design Problem (MCDP) aims at efficiently organizing the machines into cells in order to increase productivity by minimizing the inter-cell moves of parts. In this paper, we present a new approach based on Invasive Weed Optimization (IWO) for solving such a problem. The IWO algorithm is a recent metaheuristic inspired on the colonization behavior of the invasive weeds in agriculture. IWO represents the solutions as weeds that grow and produce seeds to be randomly dispersed over the search area. We additionally incorporate a binary neighbor operator in order to efficiently handle the binary nature of the problem. The experimental results demonstrate the efficiency of the proposed approach which is able to reach several global optimums for a set of 90 well-known MCDP instances.

Keywords: Manufacturing Cell Design, Invasive Weed Optimization, Metaheuristics, Optimization.

1 Introduction

The Manufacturing Cell Design Problem (MCDP) is a group technology application that consists in grouping components according to the next statement: *'Similar things should be manufactured in the same way'* [10]. The MCDP is represented through functionally diverse machines, which are grouped in cells, each of which is dedicated to the production of a part family, composed of different parts with similar processing requirements [20]. Then, the goal of the MCDP is to find machine-part's associations with the least amount of part movements between cells.

During the last decades, the MCDP has been tackled via approximate and exact methods. On the one hand, approximate methods are focused on finding an approximate solution, which is not necessarily the global optimum. Metaheuristics such as genetic algorithms [18,5,13,3], tabu search [8,1], simulated annealing [19] and particle swarm optimization [4] have intensively been used to solve this problem. On the other hand, exact methods perform a complete search within all possible solutions. Various experimental results performed by using mathematical and constraint programming can be seen in [16] and in [2,15,14,6], respectively.

Since then, the MCDP has been modeled as a set of machines and parts grouped in a matrix called Machine-Part Incidence Matrix, which determines when a part requires the service of a machine, or otherwise. All MCDP instances are resolved by manipulating the incidence matrix in a manner such that the grouping of all similar objects is possible [20]. In this paper, we solve the MCDP by using the Invasive Weed Optimization (IWO) algorithm. The IWO algorithm is a population-based metaheuristic, which simulates the colonization behavior of the invasive weeds in agriculture [17]. It represents the solutions as a finite number of weeds that grow and produce seeds depending on its fitness, that are randomly dispersed over the search area. We illustrate promising results where the global optimum is reached in several well-known MCDP instances.

This paper is organized as follows: Section 2 describes the mathematical model for the MCDP. Section 3 presents the IWO algorithm. Section 4 illustrates the experimental results, followed by conclusions and some lines of future work.

2 Manufacturing Cell Design Problem

The MCDP is defined as a production strategy which realizes a production unit division of an organization. These units form groups or families of components, also denominated as production cells [12]. The MCDP is considered as a group technology application, in where the goals are the reduction of part movements between the cells and leads to a lot of advantages such as reduction of material-handling times and cost, reduction of labors and paper works, decrease of in-process inventories, shortening of production lead time, increase of machine utilization, and others [21]. The MCDP follows the next statement: *'Similar things should be manufactured in the same way'* [10]: similar parts either by properties such as weight, manufacturing materials or required operations, must belong to the same production unit.

First, the MCDP requires the organization of the involved elements in a representative structure of the processing requirements that the production system has. In this way, the incidence matrices are created in order to summarize the necessary information. The first matrix is denominated machine-part matrix, which determines through ones and zeros the necessary machines for the production of the parts. In the matrix, the machines are represented as rows and the parts are represented as columns [10]. Table 1 shows a machine-part matrix example, which a row position with a number one means that the machine processes the part associated to the respective column. Then, the goal is the grouping of machines that process similar parts, in the same way as the example matrix showed in Table 2.

The MCDP is a model that must be satisfied for finding an optimum cell organization, which is described through a rigorous mathematical formulation of the problem as follows[16]:

Table 1: Machine-Part Matrix.

	Part										
Machine	1	2	3	4	5	6	7	8	9	10	11
A			1				1				1
B	1	1				1					
C		1				1			1		
D				1	1					1	
E			1				1				
F			1								1
G					1			1		1	

Table 2: Processed Machine-Part Matrix.

	Part										
Machine	3	7	11	1	2	6	9	4	5	8	10
A	1	1	1								
E	1	1									
F	1										
B				1	1	1					
C					1	1	1				
D								1	1		1
G									1	1	1

- M: number of machines.
- P: number of parts.
- C: number of cells.
- i: index of machines ($i = 1, 2, ..., M$).
- j: index of parts ($i = 1, 2, ..., P$).
- k: index of cells ($i = 1, 2, ..., C$).
- M_{max}: maximum number of machines per cell.
- $A = [a_{ij}]$: is the binary machine$\times$part incidence matrix, where:

$$a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ processes the part } j \\ 0 & \text{otherwise} \end{cases}$$

- $B = [b_{ik}]$ is the binary machine$\times$cell incidence matrix, where:

$$b_{ik} = \begin{cases} 1 & \text{if machine } i \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases}$$

- $C = [c_{jk}]$ is the binary part$\times$cell incidence matrix, where:

$$c_{jk} = \begin{cases} 1 & \text{if part } j \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases}$$

The objective function models the minimization of part movements among cells as depicted in Eq. 1.

$$Z = \sum_{k=1}^{C} \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij} c_{jk} (1 - b_{ik}) \tag{1}$$

The objective function is subjected to the following constraints:

$$\sum_{k=1}^{C} b_{ik} = 1 \quad \forall i \tag{2}$$

$$\sum_{k=1}^{C} c_{jk} = 1 \quad \forall j \tag{3}$$

$$\sum_{i=1}^{M} b_{ik} \leq M_{max} \quad \forall k \tag{4}$$

Eq. 2 defines that each machine belong to one and only one cell, Eq. 3 guarantees that each part is assigned to one and only one cell, and Eq. 4 determines the maximum number of machines that a cell can contain.

3 Invasive Weed Optimization Algorithm

In [11], the authors introduced the Invasive Weed Optimization (IWO) Algorithm, which is based on the colonization behavior of invasive weeds. Generally speaking, a weed is a plant that grows where it is not desired. In agriculture this term is used especially for plants whose growth habits are a threat to cultivated plants. Weeds exhibit interesting properties as for instance robustness and adaptivity [17]. The metaheuristic goal is to find the right places for the growth and reproduction of the weeds [7].

Therefore, each solution for the problem is represented by a weed [7]. IWO algorithm generates a set of weeds, which is called Initial Population. The weed with the best fitness among all others is known as Initial Solution. Therefore, each weed generates sets of solutions called seeds, through reproduction behaviors. When the IWO algorithm has generated a certain amount of weeds and seeds, a ranking is elaborated and it is ordered according to the fitness of the weeds. The worse ones are removed [17].

3.1 Initialization

The first step of IWO algorithm corresponds to the initialization. It is related with the obtaining a set of possible solutions for the problem [9]. Then, a group of weeds is generated and they are known as W, which contains an initial number of solutions denominated by the previously defined parameter P_{init} [7]. The initialization in the IWO algorithm performs an analysis of the weeds, selecting the one with the lowest fitness. The selected weed will be the initial optimum for the metaheuristic. The initialization phase is stated in Eq. 5:

$$W^i \in (U(X_{min}, X_{max})^d) \qquad (1 \leq i \leq P_{init})\,(1 \leq d \leq D) \tag{5}$$

The W^i variable is the ith solution of the W group, i.e. $W^i \in W$, and D is known as the number of dimensions or variables of the problem. X_{min} is the minimum posible value that a dimension defined by $d \in (1..D)$ can take. Further, X_{max} is the maximum posible value that the dimension can obtain.

3.2 Reproduction

The reproduction is the second step of the IWO algorithm, which refers to the generation of new solutions, that are known as seeds, from the weeds previously created in the initialization phase. The goal of the reproduction is the exploration of the search space in order to improve the fitness values of the existing weeds. For this purpose, the number of seeds S^p_{num} is calculated for each weed according to Eq. 6:

$$S^p_{num} = S_{min} + (\frac{F(W^p) - F_{worse}}{F_{best} - F_{worse}})(S_{max} - S_{min}) \qquad (1 \leq P \leq P_{init}) \tag{6}$$

The S_{min} and S_{max} parameters are the minimum and maximum number of allowed seeds per weed [7]. $F(W^p)$ is the fitness value for the evaluated weed W^p, while F_{worse} and F_{best} are the worst and the best fitness value within the set of weeds W, respectively.

3.3 Spatial Dispersal

The next procedure is to create seeds for each weed p. The set of seeds S^p is computed through the formula presented in Eq. 7:

$$(S^r_d)^p = w^p_d + \mathcal{N}(0, \theta_G)^D \qquad (1 \leq r \leq S^p_{num})\,(1 \leq d \leq D) \tag{7}$$

whereby $(S^r_d)^p$ represents the dth dimension of the rth seed for the pth weed of the W set. The w^p_d weed is moved in the neighborhood for the seed creation by using a normal distribution ($\mathcal{N}(0, \theta_G)^D$) with zero mean and varying standard deviation represented by θ_G. The standard deviation calculation is performed for each generation, represented by G, through the formula showed in Eq. 8:

$$\theta_G = \theta_{final} + \frac{(N_{iter} - G)^\theta_{mod}}{(N_{iter})^\theta_{mod}}(\theta_{init} - \theta_{final}) \tag{8}$$

whereby N_{iter} is the maximum number of iterations for the seed generation. θ_{init} and θ_{final} are previously defined parameters, and θ_{mod} denotes a non-linear modulation index [7].

3.4 Exclusive Competition

The last step of the IWO algorithm consist in a comparison between weeds and seed according to the fitness value. This process occurs when the maximum number of weeds and seeds, which is known as P_{max}, is reached. P_{max} is a previously defined parameter of the metaheuristic. After passing some iterations, the number of weeds in a colony will reach its maximum level by fast reproduction, however, it is expected that the fitter weeds have been reproduced more than the undesirable weeds. By reaching the maximum number of weeds in the colony (P_{max}), a mechanism for eliminating the weeds with poor fitness in the generation is activated [9].

The elimination mechanism is known as Exclusive Competition and works as follows: when the maximum number of weeds and seeds in a colony is reached, they are ranked together, considering the seeds as weeds now. Next, the weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, the weeds with better fitness survive and are allowed to replicate. The population control mechanism is also applied to their offspring up to the end of a given run, performing competitive exclusion [9].

3.5 Binary Invasive Weed Optimization Algorithm

In Eq. 7, the seed generation uses a normal distribution operator on its respective weed. However, this function operates with a real domain, and the MCDP has a binary domain $B^D = 0{,}1$, $(1 \leq d \leq D)$. Therefore, the function needs an adaptation for binary values, which changes the normal distribution as presented in Eq. 9:

$$(S_d^r)^p = \mathcal{N}(w_d^p, \theta_G)^D \qquad (1 \leq r \leq S_{num}^p)\ (1 \leq d \leq D) \tag{9}$$

The new function is known as Binary Neighbor Operator. As first step, the number of those bits is determined in order to obtain a new different solution represented for the seed. These numbers of bits are drawn from a normal distribution to keep a senseful standard deviation θ_G. Based on the number of bits, the probability of a single bit to be changed is computed in a second step. Finally, the given weed w^p is copied to the seed S and all D bits of this seed S are changed according to the pre-computed probability [7].

The Binary Neighbor Operator is defined through Algorithm 1, which shows the criteria for the change of each bit that will generate the new seed. Finally, the complete Binary IWO algorithm is also defined in Algorithm 2.

4 Experimental Results

We have performed a set of experiments based on 90 problem instances presented in [2]. The algorithm has been implemented in Java and launched on a Intel Core i5 4210U processor with 6 GB RAM, running Windows 8.1 Pro. The obtained results are illustrated in Table 3, where the ‘Opt’ column depicts the global optimum of the instance, ‘IWO’ the result reached by the proposed approach, and RPD represents Relative Percentage Deviation, which is computed as: $RDP = \frac{(Z - Z_{opt})}{Z_{opt}} \times 100$; where Z_{opt} is the best known optimum value and Z is the best optimum value reached by IWO. The IWO algorithm was executed using the following parameters: Generation Number $(G) = 10$; Iteration Number $(N_{iter}) = 500$; Initial number of weeds $(P_{init}) = 20$; Maximum number of seeds $(P_{max}) = 10$; Minimum number of seeds $(S_{min}) = 10$; Maximum number of seeds $(S_{max}) = 20$; $\theta_{init} = MC$; $\theta_{final} = 1$; and $\theta_{mod} = 3$.

The results are quite promising, indeed the proposed IWO algorithm is able to achieve 89 of 90 global optimums, keeping a low RPD value for the remaining instance. Such results also exhibit the robustness of the approach, which is able to reach good enough optimal values by keeping the same parameter configuration. Figures 1 and 2 depict representative convergence charts, where we can observe a fast convergence, achieving optimums before 500 iterations.

Algorithm 1 Binary Neighbor Operator

Require: : w^p, θ_G, D
$r_{bits} = \mathcal{N}^+(0, \theta_G)$
$p_{change} = \frac{r_{bits}}{D}$
$S = w^p$
for $d \in 1..D$ **do**
    $random = U(0,1)$
    if $random \leq p_{change}$ **then**
        $S_d = \neg S_d$
    end if
end for
return S

Algorithm 2 Binary IWO algorithm

Require: : $P_{init}, N_{iter}, \theta_G, S_{max}, S_{min}$
Generate initial population of weeds: W = Initialization(P_init).
for $(i = 1 : N_{iter})$ **do**
    while $(\sharp W \leq P_{max})$ **do**
        for $(p = 1 : \sharp W)$ **do**
            S^p_{num} = Reproduction(S_{max}, S_{min}, w^p).
            for $(r = 1 : S^p_{num})$ **do**
                for $(d = 1 : D)$ **do**
                    $(S^r_d)^p$ = Spatial Dispersal(w^p_d, N_{iter}, S^p_{num}, θ_G).
                end for
            end for
        end for
    end while
    W = Exclusive Competition(W, S).
end for
return w_{best}

Table 3: Experimental Results with C=2 and C=3

	C=2														
P	$M_{max} = 8$			$M_{max} = 9$			$M_{max} = 10$			$M_{max} = 11$			$M_{max} = 12$		
	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD
1	11	11	0.00	11	11	0.00	11	11	0.00	11	11	0.00	11	11	0.00
2	7	7	0.00	6	6	0.00	4	4	0.00	3	3	0.00	3	3	0.00
3	4	4	0.00	4	4	0.00	4	4	0.00	3	3	0.00	1	1	0.00
4	14	14	0.00	13	13	0.00	13	13	0.00	13	13	0.00	13	13	0.00
5	9	9	0.00	6	6	0.00	6	6	0.00	5	5	0.00	4	4	0.00
6	5	5	0.00	3	3	0.00	3	3	0.00	3	3	0.00	2	2	0.00
7	7	7	0.00	4	4	0.00	4	4	0.00	4	4	0.00	4	4	0.00
8	13	13	0.00	10	10	0.00	8	8	0.00	5	5	0.00	5	5	0.00
9	8	8	0.00	8	8	0.00	8	8	0.00	5	5	0.00	5	5	0.00
10	8	8	0.00	5	5	0.00	5	5	0.00	5	5	0.00	5	5	0.00

	C=3											
P	$M_{max} = 6$			$M_{max} = 7$			$M_{max} = 8$			$M_{max} = 9$		
	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD
1	27	27	0.00	18	18	0.00	11	11	0.00	11	11	0.00
2	7	7	0.00	6	6	0.00	6	7	16.7	6	6	0.00
3	9	9	0.00	4	4	0.00	4	4	0.00	4	4	0.00
4	27	27	0.00	18	18	0.00	14	14	0.00	13	13	0.00
5	11	11	0.00	8	8	0.00	8	8	0.00	6	6	0.00
6	6	6	0.00	4	4	0.00	4	4	0.00	3	3	0.00
7	11	11	0.00	5	5	0.00	5	5	0.00	4	4	0.00
8	14	14	0.00	11	11	0.00	11	11	0.00	10	10	0.00
9	12	12	0.00	12	12	0.00	8	8	0.00	8	8	0.00
10	10	10	0.00	8	8	0.00	8	8	0.00	5	5	0.00

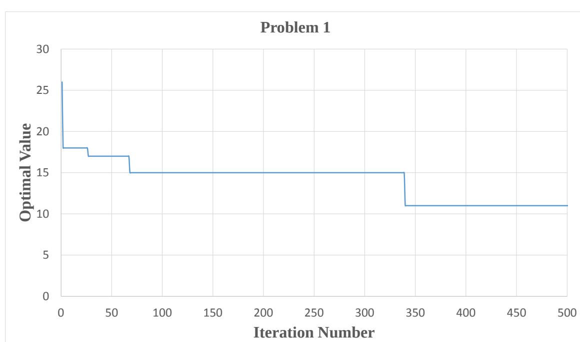

Fig. 1: Convergence charts for problem 1 with MMax=8 and C= 2.

5 Conclusions

In this paper, an invasive weed optimization algorithm for solving MCDPs was presented. A binary neighbor operator is employed to efficiently handle the binary nature of the problem. We have tested 90 well-known problem instances considering different M_{max} values and cell numbers. The results are

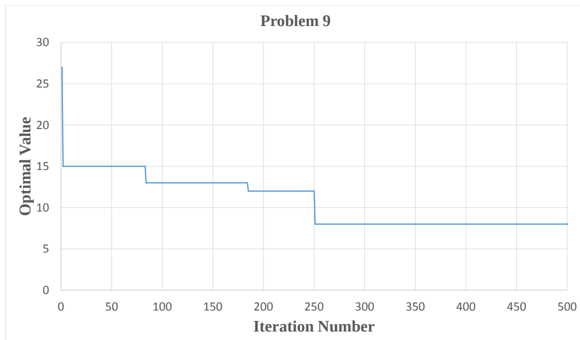

Fig. 2: Convergence charts for problem 9 with MMax=10 and C= 2.

quite promising, where the proposed algorithm is capable to achieve 89 of 90 global optimums, keeping a low RPD value for the remaining instance. Such results also exhibit the robustness of the approach, which is able to reach good enough optimal values by keeping the same parameter configuration. As future work, we plan to experiment with additional instances of the MCDP as well as to implement new modern metaheuristics for solving this problem. The study of adaptive and dynamic parameter setting to the presented approach would also be another direction for future work.

References

1. N. Aljaber, W. Baek, and C. Chen. A tabu search approach to the cell formation problem. *Computers and Industrial Engineering*, 32(1):169–185, 1997.
2. F. F. Boctor. A linear formulation of the machine-part cell formation problem. *International Journal of Production Research*, 29(2):343–356, 1991.
3. M. Boulif and K. Atif. A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. *Comput. Oper. Res.*, 33:2219–2245, 2006.
4. O. Durán, N. Rodriguez, and L. Consalter. Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. *Expert Systems with Applications*, 37(2):1563–1567, 2010.
5. T. James, E. Brown, and K. Keeling. A hybrid grouping genetic algorithm for the cell formation problem. *Computers and Operations Research*, 34(7):2059–2079, 2007.
6. A. Kusiak and W. Chow. Efficient solving of the group technology problem. *Journal of Manufacturing Systems*, 6:117–124, 1987.
7. I. Lenin, B. R. Reddy, and M. S. Kalavathi. Hybrid-invasive weed optimization particle swarm optimization algorithm for solving optimal reactive power dispatch problem. *International Journal of Research in Electronics and Communication Technology (IJRECT 2014)*, 1(1):41–45, 2014.

8. S. Lozano, A. Díaz, I. Eguía, and L. Onieva. A one-step tabu search algorithm for manufacturing cell design. *Journal of the Operational Research Society*, 50(5), 1999.
9. A. R. R. Mallahzadeh, H. Oraizi, and Z. Davoodi-Rad. Application of the invasive weed optimization technique for antenna configurations. *Progress In Electromagnetics Research*, 79:137–150, 2008.
10. P. D. Medina, E. A. Cruz, and M. Pinzon. Generacion de celdas de manufactura usando el algoritmo de ordenamiento binario (aob). *Scientia et Technica Ao XVI*, 16(44):106–110, 2010.
11. A. R. Mehrabian and C. Lucas. A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4):355–366, 2006.
12. M. Mururgan and V. Selladurai. Manufacturing cell design with reduction in setup time through genetic algorithm. *Journal of Theoretical and Applied Information Technology*, 3(1):76–97, 2006.
13. A. Nsakanda, M. Diaby, and W. Price. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research*, 171(3):1051–1070, 2006.
14. E. Olivia-Lopez and G. Purcheck. Load balancing for group technology planning and control. *International Journal of MTDR*, 19:259–268, 1979.
15. G. Purcheck. A linear - programming method for the combinatorial grouping of an incomplete set. *Journal of cybernetics*, 5:51–58, 1975.
16. R. Soto, H. Kjellerstrand, O. Durán, B. Crawford, E. Monfroy, and F. Paredes. Cell formation in group technology using constraint programming and boolean satisfiability. *Expert Systems with Applications*, 39(13):11423–11427, 2012.
17. C. Veenhuis. Binary invasive weed optimization. *Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 449–454, 2010.
18. V. Venugopal and T. Narendran. A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers & Industrial Engineering*, 22(4):469–480, 1992.
19. T. Wu, C. Chang, and S. Chung. A simulated annealing algorithm for manufacturing cell formation problems. *Expert Systems with Applications*, 34(3):1609–1617, 2008.
20. A. R. Xambre and P. M. Vilarinho. A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European Journal of Operational Research*, 151(2):434–446, 2003.
21. Y. Yin and K. Yasuda. Manufacturing cells design in consideration of various production. *International Journal of Production Research*, 40(4):885–906, 2002.